REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER NAVENVPREDRSCHFAC Contractor Report CR 82-13	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Tropical Cyclone Satellite Images in Lagrangian Coordinates		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Vern L. Peterson and Susan L. Uhrich		8. CONTRACT OR GRANT NUMBER(s) N66856-81MD-00021
9. PERFORMING ORGANIZATION NAME AND ADDRESS Ocean Data Systems, Inc. 2400 Garden Road Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE 62759N PN WF59-551 NEPRF WU 6.2-17
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Air Systems Command Department of the Navy Washington, DC 20361		12. REPORT DATE September 1982
		13. NUMBER OF PAGES 72
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Environmental Prediction Research Facility Monterey, CA 93940		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Tropical cyclones
Lagrangian coordinates
Satellite cloud images

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The purpose of the work reported here was to produce a videotape of Hurricane Ignacio (1979) that resembled time-lapse photography, except that the successive images were in the Lagrangian frame instead of the usual Eulerian frame. Procedures and programs are described for taking the original satellite images of the storm, finding the storm centers, rotating the images the proper amount, and displaying them in succession for copying to videotape using special hardware. ((continued on reverse))

Abstract, Block 20, continued.

The procedures and programs for replacing missing or garbled images are also described; these are based upon a cross-correlation analysis of the "before" and "after" images, followed by a linear interpolation of the properly rotated images. Recommendations for future work are given. Appendices contain job control streams, flow charts, and listings of all the software developed for this project.

The videotape developed in this project has been delivered separately. It contains first the sequence of original images (with the storm properly centered), and then the sequence of Lagrangian images.

AN (1) AD-A121 130
FG (2) 040200
FG (2) 120500
CI (3) (U)
CA (5) OCEAN DATA SYSTEMS INC MONTEREY CALIF
TI (6) Tropical Cyclone Satellite Images in Lagrangian
Coordinates.
TC (8) (U)
DN (9) Final rept.,
AU (10) Peterson,Vern L.
AU (10) Uhrich,Susan L.
RD (11) Sep 1982
PG (12) 71p
CT (15) N66856-81-MD-00021
PJ (16) F59551
RN (18) NEPRF-CR-82-13
RC (20) Unclassified report
DE (23) *Hurricanes, *Image processing, Aerial photography,
Artificial satellites, Photographic images, Image
processing, Sequences, Clouds, MOtion, Cross
correlation, Interpolation, Rotation, Video tape
recording, Computer programs
DC (24) (U)
ID (25) Hurricane Ignacio, Satellite photography, Lagrangian
coordinates, PE62759N, WU6217
IC (26) (U)
AB (27) The purpose of the work reported here was to produce a videotape of Hurricane Ignacio (1979) that resembled time-lapse photography, except that the successive images were in the Lagrangian frame instead of the usual Eulerian frame. Procedures and programs are described for taking the original satellite images of the storm, finding the storm, centers, rotating the images the proper amount, and displaying them in succession for copying to videotape using special hardware. (Author)
AC (28) (U)
DL (33) 01
SE (34) F
CC (35) 408787

#

ROUTINE REPLY, ENDORSEMENT, TRANSMITTAL OR INFORMATION SHEET

OPNAV 5216/158 (Rev. 7-78)
SN 0107 LF 052-1691

A WINDOW ENVELOPE MAY BE USED
Formerly NAVEXOS 3789

CLASSIFICATION (*UNCLASSIFIED when detached from enclosures, unless otherwise indicated*) UNCLASSIFIED

FROM (*Show telephone number in addition to address*)
Commanding Officer, Naval Environmental Prediction Research Facility, Monterey, CA 93940 AVN 878-2928

DATE
25 Oct 82

SUBJECT
NAVENVPREDRSCHFAC technical publication; forwarding of

SERIAL OR FILE NO.
5600 NEPRF/SBB:sb
SER 358

TO:

Distribution
[see p. Dist-1, encl. (1)]

REFERENCE

ENCLOSURE
(1) NAVENVPREDRSCHFAC Contractor Report CR 82-13: Tropical Cyclone Satellite Images in Lagrangian Coordinates

VIA:

ENDORSEMENT ON

XX	FORWARDED
	RETURNED
	FOLLOW-UP, OR TRACER
	REQUEST
	SUBMIT
	CERTIFY
	MAIL
	FILE

	GENERAL ADMINISTRATION
	FOR APPROPRIATE ACTION
	UNDER YOUR COGNIZANCE
XX	INFORMATION & retention
	APPROVAL RECOMMENDED ☐ YES ☐ NO
	☐ APPROVED ☐ DISAPPROVED
	COMMENT AND/OR CONCURRENCE
	CONCUR
	LOANED, RETURN BY:
	SIGN RECEIPT & RETURN
	REPLY TO THE ABOVE BY:
	REFERENCE NOT RECEIVED
	SUBJECT DOCUMENT FORWARDED TO:
	SUBJECT DOCUMENT RETURNED FOR:
	SUBJECT DOCUMENT HAS BEEN REQUESTED, AND WILL BE FORWARDED WHEN RECEIVED
	COPY OF THIS CORRESPONDENCE WITH YOUR REPLY
	ENCLOSURE NOT RECEIVED
	ENCLOSURE FORWARDED AS REQUESTED
	ENCLOSURE RETURNED FOR CORRECTION AS INDICATED
	CORRECTED ENCLOSURE AS REQUESTED
	REMOVE FROM DISTRIBUTION LIST
	REDUCE DISTRIBUTION AMOUNT TO:

	CONTRACT ADMINISTRATION
	NAME & LOCATION OF SUPPLIER OF SUBJECT ITEMS
	SUBCONTRACT NO. OF SUBJECT ITEM
	APPROPRIATION SYMBOL, SUBHEAD, AND CHARGEABLE ACTIVITY
	SHIPPING AT GOVERNMENT EXPENSE ☐ YES ☐ NO
	A CERTIFICATE, VICE BILL OF LADING
	COPIES OF CHANGE ORDERS, AMENDMENT OR MODIFICATION
	CHANGE NOTICE TO SUPPLIER
	STATUS OF MATERIAL ON PURCHASE DOCUMENT

	PERSONNEL
	REPORTED TO THIS COMMAND:
	DETACHED FROM THIS COMMAND
	OTHER

REMARKS (*Continue on reverse*)

Enclosure (1) describes procedures for producing a satellite image video tape production. The life cycle of Hurricane Ignacio (1979) in Lagrangian coordinates is illustrated in the production.

SIGNATURE & TITLE
Kenneth L. Van Sickle
Commanding Officer

COPY TO:

CLASSIFICATION (*UNCLASSIFIED when detached from enclosures, unless otherwise indicated*) UNCLASSIFIED

★ G.P.O.: 1979-689-016/4281

NAVENVPREDRSCHFAC
CONTRACTOR REPORT
CR 82-13

NAVENVPREDRSCHFAC CR 82-13

TROPICAL CYCLONE SATELLITE IMAGES IN LAGRANGIAN COORDINATES

Prepared By:

Vern L. Peterson and Susan L. Uhrich
Ocean Data Systems, Inc.
Monterey, CA 93940

Contract No. N66856-81MD-00021

SEPTEMBER 1982

Prepared For:
NAVAL ENVIRONMENTAL PREDICTION RESEARCH FACILITY
MONTEREY, CALIFORNIA 93940

QUALIFIED REQUESTORS MAY OBTAIN ADDITIONAL COPIES FROM THE DEFENSE TECHNICAL INFORMATION CENTER. ALL OTHERS SHOULD APPLY TO THE NATIONAL TECHNICAL INFORMATION SERVICE.

TABLE OF CONTENTS

1.0 INTRODUCTION

Most studies of cloud motions and morphology have been done in the Eulerian frame, that is, a frame fixed with respect to the earth. For most such studies, the Eulerian frame is indeed the most meaningful. For tropical cyclones, however, it would also be useful to study cloud motions and morphology in the Lagrangian frame, that is, a frame centered on the storm (and therefore moving with it) and oriented with a principal axis in the direction of motion of the storm. Such studies would hopefully reveal new insights into the behaviour of tropical cyclones.

With the advent of cloud imagery from the Geostationary Operational Environmental Satellites (GOES) and similiar satellites, this type of study is now possible, for it requires a long series of images taken at regular intervals, preferably intervals as short as possible. With GOES, the images can be generated every half hour for images of the entire disc of the earth, or every 15 minutes if only a portion of the full disc is viewed. From these images subsets can be derived that are centered on the storm.

The purpose of the contract discussed in this report was to develop software that would transform tropical cyclone images (from GOES) into Lagrangian coordinates, create interpolated images to replace missing or garbled ones, and transfer the resulting images to videotape for later playback through a normal TV monitor. The images provided by the government were of Hurricane Ignacio, which occurred off the west coast of Mexico during October 1979. The images provided were spaced two hours apart and covered a 6.5 day period. In all, 76 good images were provided; three new images had to be generated by interpolation to replace two missing and one garbled image. Thus, a total of 79 images in the Lagrangian frame were produced and transferred to videotape.

In the next section, we discuss the software (including procedures for using it) and production of the videotape. The third section discusses the results. Finally, Appendices A, B and C present flow charts and listings of the software. A list of the abbreviations used in this report is contained in the final appendix (D).

2.0 IMAGE PROCESSING

The transformation of an image of a tropical cyclone into Lagrangian coordinates and final storage onto videotape is a process that has many steps. The process involves the use of the Data General Computers associated with SPADS (either the Eclipse S/250 or the Nova), the CDC computers of FNOC (SPC and, possibly, HAL), and a special piece of hardware known as an RBG to NTSC color encoder. The entire process is summarized in the flow chart shown in Figure 2-1. The following sections discuss the software referred to in this figure as well as the procedures for using it and for transferring the images to videotape.

2.1 Initial Processing on SPADS

The initial processing of the image uses software previously developed for the SPADS. This software is documented elsewhere and will only be briefly mentioned here. The exception to this is the program ANGLEL, which was developed specifically for this contract; it is discussed toward the end of this section.

The user must start with a large sector image file in a format compatible with the SPADS. This image file can either be directly captured from GOES using GADHS or it can be taken from magnetic tape in the MCIDAS format and converted to the format acceptable to SPADS using the program MCODIS or MCODIS2. This image sector is then subset into a 512 x 512 image using GIMAGE; the starting location (upper left corner) of this subset is chosen such that the center of the tropical cyclone is close to the center of the image. (It may be necessary to run GIMAGE several times on a given sector in order to properly center the storm.) The resultant image is then processed with the program TCIF (for Tropical Cyclone Intensity Forecast) to better estimate the storm center location (in both pixel/raster and latitude/longitude coordinates). The program ZEBFLS is then run to transfer the image produced by GIMAGE to magnetic tape in a format acceptable to the FNOC computer SPC. Finally, ANGLEL is run to determine the amount of rotation needed for each image in the Lagrangian transformation.

ORDER OF EVENTS

ON SPADS

1. Generate large sector image files in format acceptable to SPADS

2. Subset sector image files to generate 512 x 512 files centered on storm

3. Find coordinates of spiral centers in both (pixel, raster) and (latitude, longitude)

4. Store 512 x 512 images on tape in format acceptable to SPC

5. Calculate rotation angle needed for each image

ON SPC

6. Rotate each image and save new images on tape

7. Replace missing images (if any) using interpolation scheme and save on tape

ON SPADS

8. Read new images from tape to disk

9. Display new images chronologically and transfer to videotape using RBG to NTSC color encoder

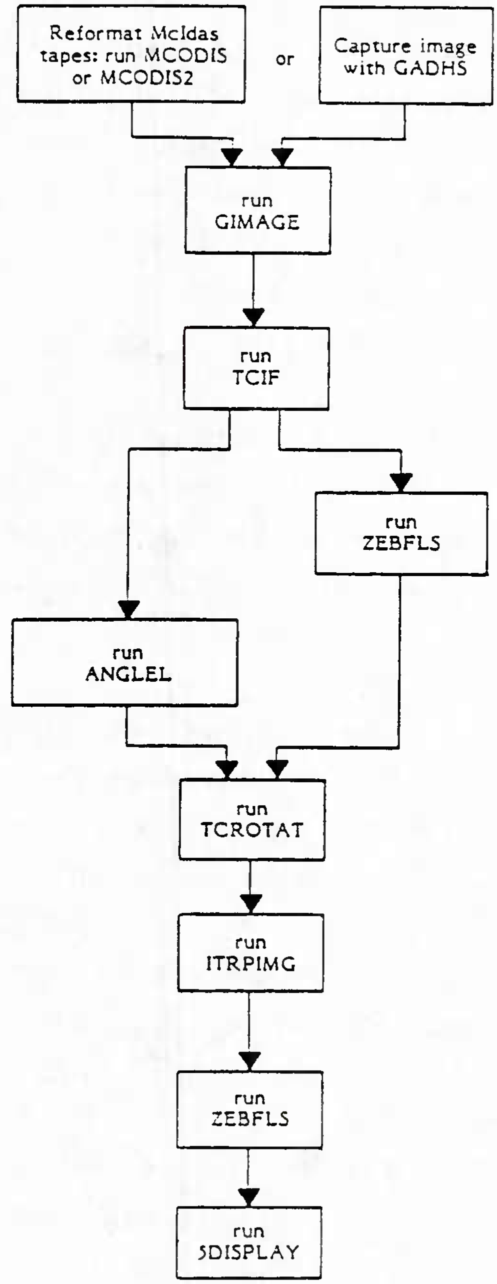


FIGURE 2-1: SUMMARY OF IMAGE TRANSFORMATION PROCESS

The program ANGLEL and its subroutine CKIN are listed in Appendix A along with their flow charts. This program calculates the angle that an image must be rotated in order to put it into the Lagrangian frame. The center of the storm (determined using TCIF) is the origin of the Lagrangian frame and the vertical axis becomes the direction of its motion. If two consecutive images are denoted by the subscripts i and i + 1 then the change in latitude and longitude of the storm center is

$$\Delta \text{ lat} = \text{lat}_{i+1} - \text{lat}_i$$
$$\Delta \text{ long} = \text{long}_{i+1} - \text{long}_i$$

The angle of rotation, θ , needed to put the image into the Lagrangian frame is then given by

$$\tan \Theta = \tan\left(\cos(\text{lat}) \cdot \Delta \text{ long}\right)/\sin(\Delta \text{lat})$$

where lat is the higher latitude of the two points. In order to speed computations, this equation was approximated to

$$\tan \theta = \cos(\text{lat})\,(\Delta \text{ long}/\Delta \text{ lat})$$

since Δ lat and Δ long are such small angles. (The inaccuracy introduced by this approximation is considerably smaller than the inaccuracies of the Δ lat and Δ long measurements caused by errors in storm center positioning.) If the second point (i+1) is equatorward of the first point (i) then θ is adjusted by adding or subtracting 180°. The results of using this program are discussed in section 3.0.

2.2 Processing on SPC

The images stored on tape by ZEBFLS can now be processed with the SPC computer at FNOC. As indicated in Figure 2-1, two programs are used in this processing. The first is TCROTAT, which reads an image from tape, rotates it through the angle calculated by ANGLEL, and writes the new image to another tape. The second is ITRPIMG, which creates an interpolated image from two other images (the "before" and "after" images that had already been rotated) and writes the resultant image to tape. These programs are executed with the job card streams JTCROTATE and JINTERPOLATE. These programs will now be discussed in detail.

2.2.1 Program TCROTAT

The job stream listing, flow charts and program and subroutine listings of TCROTAT are given in Appendix B. The input data to this program consists of the storm center location in (pixel, raster) coordinates (as found by running TCIF) and the angle the image needs to be rotated (as found by running ANGLEL). Columns 1-3 of the input record contain ICNTR, the pixel value of the storm center; columns 5-7 contain JCNTR, the raster value of the center; and columns 9-15 contain ANGLE, the rotation angle for the image (positive counterclockwise). (Warning: Do not invert pixel and raster number when using TCROTAT. This is easy to do since in TCROTAT the pixel is labeled I and the raster J whereas in TCIF the notation is reversed.) The user also has the option of using the line printer to display the central portion (128 x 128) of the storm for both the original and rotated images; this option is exercised through the setting of switches in the job control stream. The program thus starts by reading the switches and the input data (storm center and rotation angle) and checking for validity.

The image file is then read from the tape (referred to as TAPE4 in the program) and stored in Extended Core Storage (ECS). An image file consists of 515 records: record 1 is the header, records 2-513 are image data (one raster per record), and records 514 and 515 are documentation. Each raster (or record) contains 512 8-bit pixels (or bytes) packed into 69 words; the image file therefore is 69 x 515 = 35535 words long. As the image is read in, it is shifted such that the spiral center location will be at pixel 256 and raster 256. This means that blank rows and columns will be inserted where the image has moved away from the edges. The first word address (FWA) of each raster is stored in IEC1(1) through IEC1(512), the FWA of the header record is stored in IEC1(513) and the FWA's of the documentation are stored in IEC1(514) and IEC1(515). This array is contained in the COMMON block ECBLK, along with a similar array for the rotated image. This second array, IEC2, is 512 words long, having only the FWA's of the rasters of the rotated image since the header and documentation records will be the same as for the original image.

If switch 1 is set, the central portion (128 x 128) of the original image (shifted to the storm center) will be displayed, using the line printer as the display devise. Since the line printer outputs characters at 10 per inch horizontally and 6 or 8 per inch vertically, the resultant image is elongated vertically. This distortion is of no consequence since these intermediate displays are only for verification of storm center and rotation.

The line printer display is executed with a call to the subroutine IMGPLOT, which is passed the argument ITYP. If ITYP = 1, the original image is displayed; if ITYP = 2, the rotated image is displayed on the line printer. This subroutine reads the middle 128 rasters of the image from ECS. It unpacks the data, assigns a printer character for each pixel intensity according to Table 2-1, and prints the line. As stated above, only the central 128 x 128 portion of the image is displayed.

The image is then rotated with a call to subroutine ROTATE, which is passed the argument ANGLE. The latter is the angle, in degrees, the image is to be rotated, with positive angles being counterclockwise. In order to save computer time the corners of the 512 x 512 image are not rotated, since they would be rotated out of the image and be lost anyway. This is accomplished by calculating the start and end points for each raster

$$\text{END PT} = \left(255^2 - (Y - 255)^2\right)^{1/2} + 256$$

$$\text{START PT} = 512 - \text{END PT}$$

where

$$Y = \text{raster number} - 1$$

If (x,y) is the location of a pixel in the original image, then the location (x',y') of that pixel in the rotated image can be calculated from

$$x' = x \cdot \cos(\text{ANGLE}) + y \cdot \sin(\text{ANGLE})$$

$$y' = -x \cdot \sin(\text{ANGLE}) + y \cdot \cos(\text{ANGLE})$$

If I is the pixel number (I = 1 at the left edge of the screen) then x = I - 256.

TABLE 2-1

CORRESPONDENCE BETWEEN SATELLITE IMAGE INTENSITY RANGES AND LINE PRINTER DISPLAY CHARACTERS

INTENSITY RANGE	CHARACTER
0-15	BLANK
16-31	. (PERIOD)
32-47	: (COLON)
48-63	- (HYPHEN)
64-79	+
80-95	*
96-111	1
112-127	2
128-143	3
144-159	4
160-175	5
176-191	6
192-207	7
208-223	8
224-239	9
240-255	≡

Likewise, if J is the raster number (J = 1 at the top of the screen), then y = 256 - J. The center of the screen is therefore at (x, y) = (o, o) and (I, J) = (256, 256).

The coordinates (x, y) of all the pixels in the original image are integers but the coordinates (x', y') of the pixels in the rotated image are, in general, not integers. A bilinear interpolation was therefore performed to assign the proper intensities to the integer pixel location of the new image. If x' of the new image falls between x_i and x_{i+1} of the original image, and if y' falls between y_j and y_{j+1}, then

$$I(x', y') = I_o + (I_1 - I_0)\ \Delta x + (I_2 - I_0)\ \Delta y + (I_3 + I_0 - I_2 - I_1)\,\Delta x\,\Delta y$$

where

$$\begin{aligned} I_0 &= I\,(x_i, y_j) \\ I_1 &= I\,(x_{i+1}, y_j) \\ I_2 &= I\,(x_i, y_{j+1}) \\ I_3 &= I\,(x_{i+1}, y_{j+1}) \\ \Delta x &= x' - x_i \\ \Delta y &= y' - y_j \end{aligned}$$

In practice, the (x', y') are kept integer and the corresponding (x, y) locations calculated, then the above interpolation equation is used to assign the proper intensity to the pixel at (x', y').

In order to minimize central memory (CM) requirements, the new image is built one raster at a time, bringing into CM only those portions of the original image that are necessary. The new image is packed, in ECS, in the same format as the original image.

Upon completion of image rotation, switch 2 is tested. If it is ON then IMGPLOT is called again, this time to display the central portion of the rotated image.

Finally, the rotated image is written to TAPE3 along with the same header and documentation as on the original image. The output tape has a format identical to that of input tape as it will be read by SPADS.

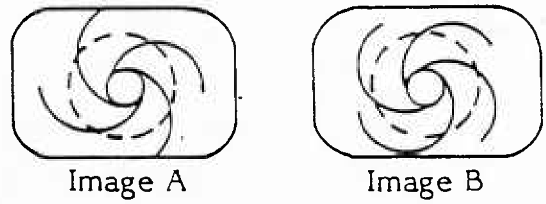
In using JTCROTATE, the user must take care to change the number of files skipped on each tape as well as use the proper VSN (volume serial number) for each tape, each time this job stream is executed.

2.2.2 Program ITRPIMG

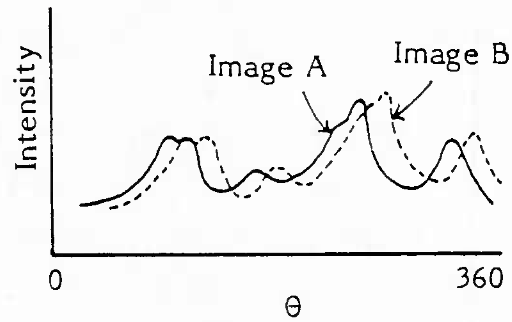
The job stream listings, flow charts, and program and subroutine listings of ITRPIMG are given in Appendix C. This program replaces missing or garbled images by interpolating between "before" and "after" images. Since the "before" and "after" images generally have different orientations they must be rotated slightly relative to one another before the interpolation is done. The amount of rotation needed is determined by a cross-correlation calculation, in which the pixel intensities on a circle of arbitrary radius (input by the user) on one image are multiplied by the corresponding pixel intensities of the other image. The sum of these products is a measure of the degree of cross-correlation between the two images. By rotating these circles relative to each other and repeating the cross-correlation calculation for each relative orientation, the "best" orientation is determined by noting where the cross correlation is a maximum. This process is shown schematically in Figure 2-2.

The program starts by reading switch 1 and the input "card", which contains the user-supplied radius (I3 format) of the circle used for the cross-correlation calculation. This radius, which is measured in the unit corresponding to the size of the pixel, must be positive and less than 255. The program verifies this, then reads in the "before" and "after" images with a call to the subroutine READIMG; the "before" image is read from TAPE3 and the "after" from TAPE4. (In the job stream JINTERPOLATE be sure the correct VSN is specified and that the proper files have been copied to TAPE3 and TAPE4 prior to executing ITRPRMG.)

1. Start with "before" and "after" images. User specifies radius of circle for cross-correlation.



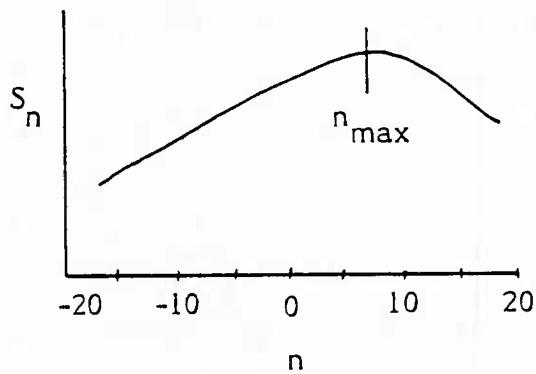
2. Determine intensities around circumference of each cross-correlation circle, interpolating where necessary. The step size, $\Delta\theta$, corresponds to one pixel diameter.



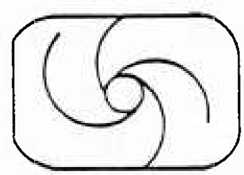
3. Calculate cross-correlation between $I_A(\theta)$ and $I_B(\theta)$

$$S_n = \sum_i I_A(\theta_i)\, I_B(\theta_i + n\,\Delta\theta)$$

for various values of $n\,\Delta\theta$ and pick value of n giving highest S_n.



4. Calculate interpolated image by rotating image A by ½ $n_{max}\,\Delta\theta$ and image B by -½ $n_{max}\,\Delta\theta$ and averaging each pixel.



Interpolated Image

FIGURE 2-2: INTERPOLATION PROCESS

The images are stored in ECS. The first word addresses (FWA) are stored in IEC1 and IEC2. IEC1 being the FWA's of the before image and IEC2 being the FWA's of the after image. The images are stored in the same format as in TCROTAT.

If switch 1 is ON, the central portions (128 x 128 pixels) of each image ("before" and "after") are displayed, using the line printer, by a call to the subroutine PRT. This subroutine is almost identical to the subroutine IMGPLOT, which is associated with the program TCROTAT.

The program then determines the number of pixels that would fit on the cross-correlation circle (this equals the circumference) and calculates the angular displacement, $\Delta\theta$, that corresponds to the width of one pixel at the distance r, the radius of the circle. The (x,y) coordinates of each pixel on the circle in steps of $\Delta\theta$ are then determined and the corresponding pixel intensities found with a bilinear interpolation; this is done for both images. The cross-correlation of the resulting one-dimensional intensity arrays is then given by

$$S_n = \sum_i I_A(\Theta_i) I_B(\Theta_i + n\,\Delta\Theta)$$

where the subscripts A and B refer to the two images, i is the pixel number, and n is the measure of the amount one array is shifted relative to the other. The maximum correlation corresponds to the largest S_n. To save computer time, the S_n are first calculated from n = -25 to n = +25 in steps of Δ n=5. Then n_m, the value of n for the maximum S_n from these coarse resolution calculations, is used as the mid-point of fine resolution (Δ n=1) calculations of S_n over the interval $n_m \pm 5$. Then n_{max}, the value of n for the greatest S_n within this range, is determined. The needed rotation of one image relative to the other to achieve maximum correlation is then n_{max} .

The above mentioned bilinear interpolation is performed with a call to subroutine INTRP. The arguments passed to this subroutine by the calling program are IFILE, the number of the image file (1 for "before" or 2 for "after") and the pixel and raster numbers (x,y). The subroutine then reads from ECS the raster on either side of y and calculates the intensity at pixel location x with a bilinear interpolation; this intensity is returned to the calling program through the argument INT.

The cross-correlation calculation and determination of the n_m or n_{max} is performed with a call to subroutine SUMSN. The arguments passed to this subroutine are NOPTS, the number of points (pixels) on the cross-correlation circle and NINC, the increment in n to use. The subroutine returns the argument NMAX, the value of n the corresponds to the maximum S_n for this NINC.

The program then rotates each image by half of the total required: the first image is rotated $\frac{1}{2}\, n_{max}\, \Delta\Theta$ and the second $-\frac{1}{2}\, n_{max}\, \Delta\Theta$. The rotated images are then averaged, pixel by pixel, to generate the final image, which is written to ECS. These calculations are performed through a call to subroutine ROTAVG, which is passed the argument ANGLE (= $\frac{1}{2}\, n_{max} \Delta\Theta$).

Switch 1 is then tested again and if found to be ON, the central portion of the new image is displayed with the line printer with a call to subroutine PRT, described earlier.

Finally, the new image is written to TAPE5 with a call to subroutine RITIMG. The header record is first output; it is generated as "IMAGE INTERPOLATED AFTER" followed by the first 15 characters taken from the header of the "before" image. The rasters are then read from ECS one at a time and output to TAPE5. Finally, the two documentation records from the "before" image are copied to TAPE5. This file is written to magnetic tape in the job stream following execution of the program. (Be sure the proper VSN is used and that the proper number of files have been skipped prior to copying TAPE5 to magnetic tape.)

2.3 Final Processing on SPADS

After all of the rotated images have been generated and saved on tape at SPC, they are ready for display on the SPADS and transfer to videotape. The images are transferred to videotape in chronological order with a one second view time of each frame; on playback this gives the appearance of time-lapse photography. Both the original images (which are unrotated and not perfectly centered) and the new ones (those in the Lagrangian frame) have been saved on videotape. In section 3 we discuss these results.

The hardware needed for this process is an RBG to NTSC color encoder. The Genisco display unit outputs digital red-blue-green (RBG) signals but since TV monitors and video tape recorders need to receive NTSC type signals, there is a need to convert the signals from one type to the other. ODSI personnel used the LENCO CCE-850 unit for this project. This unit is ODSI property, so the government will have to obtain its own LENCO unit if it wishes to repeat this process on another cyclone, or follow the alternative procedure outlined at the end of this section.

A schematic diagram for the attachment of the LENCO unit to the Genisco image display processor is shown in Figure 2-3. As can be seen, the attachment is quite simple. Not shown in the figure is the optional attachment of a color TV monitor to the video tape recorder. Even though the use of the TV monitor is not absolutely necessary, it is advisable to use one during the recording session so that the color adjustments on the LENCO Unit can be optimized.

One additional hardware modification is needed to complete this process. Since the Genisco image display controller normally outputs image frames at the rate of 40 per second, and the TV monitor and video recorder need an input rate of 30 frames per second, a serious synchronization problem arises. To correct this, the video output board in slot 3 of the PGP unit must be replaced with one that outputs the needed 30 frames per second. The proper board was obtained on loan from Genisco for the recording session. The high resolution RBG color monitor becomes useless when this change is made, as it needs input at a rate of 40 frames per second. This is one more reason for using a TV monitor during the recording session.

The SPADS program 5DISPLAY was used for the recording session; any of a number of other routines could have been used, however. The procedure is to first load the entire sequence of images from tape onto disc. For the original images, this is accomplished with the LOAD command but for the new images the program ZEBFLS must be used. The videotape first is allowed to record several seconds of blank image. The tape is then stopped with the PAUSE button. Images are then brought to the screen in chronological sequence and a one-second recording of each image is made, again using the PAUSE button. It is

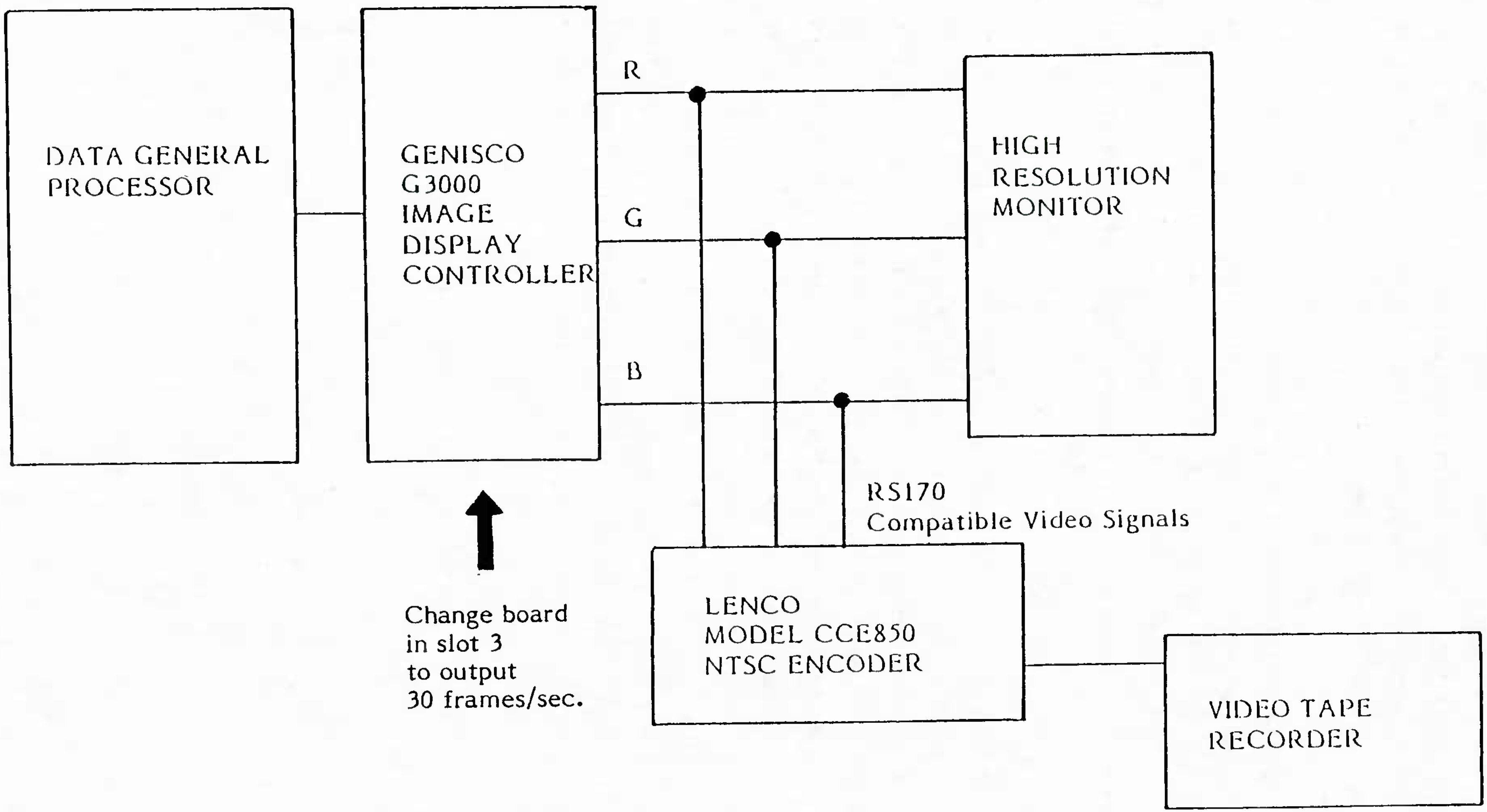


FIGURE 2-3: HARDWARE CONFIGURATION FOR DISPLAYING SATELLITE IMAGES ON VIDEO TAPE

important to use the PAUSE button instead of RECORD and STOP since in the latter case the tape motor has to come up to speed before recording can start, which causes blanks to occur in the sequence; this also makes it difficult to estimate the proper length of time to record.

Following the sequence of original images, several seconds of blank image were recorded, followed by the sequence of new images in the Lagrangian frame. In using 5DISPLAY we chose to color slice in the temperature intervals 0^o, -12^o, -24^o, -36^o, -48^o, -60^o and -72^oC. The government may wish to repeat the recording with another temperature set in order to emphasize different cloud features.

An alternative procedure to using the LENCO unit is to photograph each image on the high resolution color monitor. Color slides can then be copied to videotape using the services of a commercial photography company such as FOTOMAT. (Last year they were offering this service at a rate of $9.00 per 100 35mm slides). With this procedure, care must be taken to not disturb the camera when changing film. An advantage of this technique is that a label with the day and hour (such as IGNACIO296.18) could be affixed to the screen prior to photographing the image, thus providing a "clock" on the image. Also, titles could be inserted to introduce the sequence. A disadvantage is that film processors take little care in framing slides so that the images will probably not be properly centered. With custom processing, however, this may not be a problem.

3.0 DISCUSSION OF RESULTS

The images of Hurricane Ignacio, as captured in sequence on videotape in both the original form and in Lagrangian coordinates, reveal a number of interesting meteorological features in cloud morphology and motions. Study of these sequences also reveals some problems that should be corrected in future work; these problems could not be corrected in this study due to time limitations, as explained below.

3.1 Motions in the Eulerian Frame

The Government, during work on another project, used program GIMAGE to subset the large sector image files of Hurricane Ignacio and produce approximately centered images of this storm that were saved on tape. The government also ran program TCIF (for another project) to determine more accurate locations of the storm centers. These images and centers were provided to ODSI as the basic data set for this contract. The storm track, based on these data, is shown in Figure 3-1; the gaps marked "M" denote missing or garbled images. These images have been transferred to videotape in chronological sequence, as described above. The resulting video sequence is somewhat "jumpy" since these images are not perfectly centered (something that is difficult to do with GIMAGE), but nevertheless the sequence shows some interesting features.

The most obvious motion that can be seen in this first sequence is the counter-clockwise rotation of the central portion of the storm throughout its life. The central dense overcast (CDO) portion of the storm consists of convective clouds with great vertical development. Although motions within the CDO cannot be discerned with the temperature slicing used in this sequence, the motions along the edges of the CDO are obvious. Large convective cells are seen to form and dissolve at a distance of 100 to 200 miles from the storm center. These cells, which are sometimes larger than the CDO, move counter-clockwise about the CDO very slowly; at times they seem to compete with the CDO for becoming the new center of the cyclone. The outer spiral banded structure (OSBS) sometimes rotates about the CDO and at other times remains stationary.

HURRICANE IGNACIO, 1979
STORM TRACK

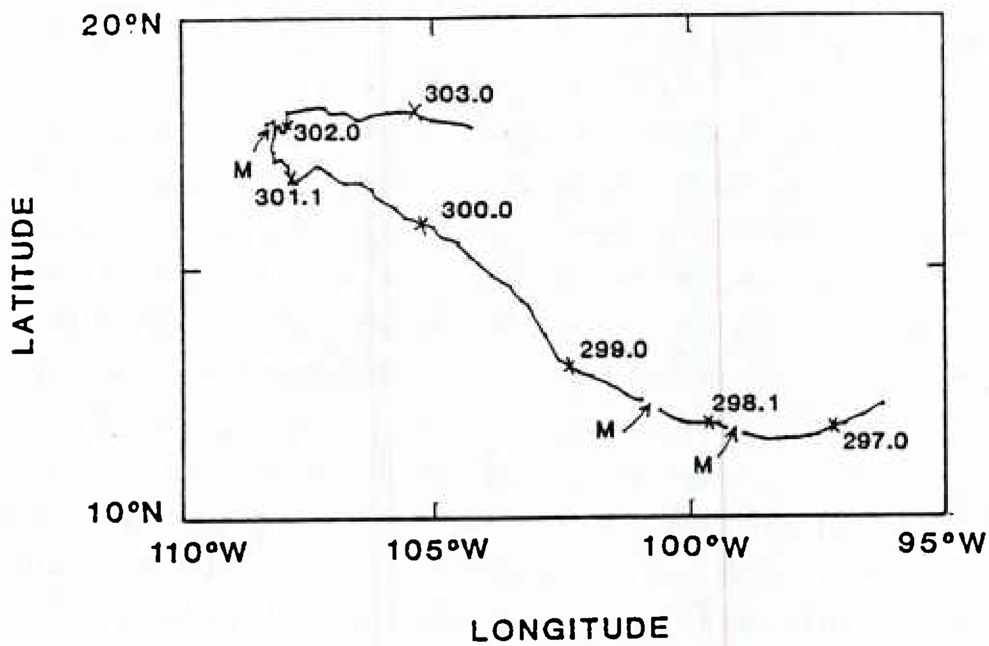


FIGURE 3-1: THE STORM TRACK OF HURRICANE IGNACIO

The gaps marked M indicate missing or garbled images. The X's mark the position of the storm at the beginning of each day; the corresponding numbers indicate the day and hour, where the day number is counted from January 1, 1979.

The size of the storm varied considerably during its life. At the start of the sequence the storm fills the entire screen; that is, it had a diameter of over 1000 miles; the CDO had a diameter of perhaps one-tenth this size. In the middle of the sequence the cyclone had shrunk to half of its original size and had a well-developed "bullseye" type central eye. Near the end of the sequence the storm had grown again and nearly attained its original size. During the period of its smaller size, the OSBS was very sharply defined but at the other times, it was more patchy.

3.2 Motion in the Lagrangian Frame

Program ANGLEL was used to calculate the rotation angles needed to produce images in the Lagrangian frame. The results are given in Table 3-1 and shown in Figure 3-2. These angles were used as input data for program TCROTAT for producing the second sequence of images seen on the videotape.

Samples of the images generated by TCROTAT are shown in Figures 3-3 and 3-5. A sample of the image interpolated between those two images is shown in Figure 3-4; the latter was generated with program ITRPIMG. It was originally thought that the image of Ignacio for day 298, hour 12, (IGNACIO 298.12) was unavailable due to a bad sector scan number (that is why the interpolated image shown in Figure 3-4 was generated), but it was later discovered that IGNACIO 298.12 was readable; it is showm in Figure 3-6. The image in the figure has been rotated by hand when photocopying so that its orientation is the same as that of the interpolated image. Comparison of Figures 3-4 and 3-6 shows that image generated by ITRPIMG is very good approximation to the true image. the corresponding cross-corrrelation function is shown in Figure 3-7; its peak is broad but smooth, and the maximum is well defined.

These Lagrangian images show features not obviously in the first sequence. Perhaps the most notable is the tendency for the large, well developed convective cells to lie in the direction of motion of the storm, or somewhat to the right of this direction; another aspect of this observation is that these convective cells do not seem to rotate about the CDO, as they did in the first sequence. Early in the life of the storm rotation is absent in the Lagrangian frame. Later, however, a pronounced rotation does occur. Late in the storm a large convective cell forms in the OSBS; significantly, this cell also lies ahead of the CDO, that is, in the direction of motion of the storm.

TABLE 3-1: LAGRANGIAN ROTATION ANGLES

TIME DAY HR	ANGLE TO ROTATE	LATITUDE	LONGITUDE
296.18	-112.71	12.15	-96.24
296.20	-109.42	11.97	-96.68
296.22	-110.63	11.87	-96.97
297.00	-105.24	11.80	-97.16
297.02	-107.88	11.76	-97.31
297.04	-111.26	11.70	-97.50
297.06	-97.27	11.62	-97.71
297.08	-86.56	11.59	-97.95
297.10	-103.66	11.60	-98.12
297.12	-85.14	11.55	-98.33
297.14	-73.28	11.57	-98.57
297.16	-66.94	11.67	-98.91
297.20	-62.93	11.82	-99.27
297.22	-90.00	11.92	-99.47
298.01	-90.00	11.92	-99.60
298.02	-92.25	11.92	-99.81
298.04	-77.65	11.91	-100.07
298.06	-66.34	11.94	-100.21
298.08	-69.02	12.00	-100.35
298.10	-59.04	12.12	-100.67
298.14	-70.61	12.29	-100.96
298.16	-52.17	12.40	-101.28
298.18	-64.43	12.65	-101.61
298.20	-62.84	12.79	-101.91
298.22	-60.76	12.92	-102.17
299.00	-55.60	13.04	-102.39
299.02	-28.92	13.18	-102.60
299.04	-28.21	13.69	-102.89
299.06	-39.71	14.16	-103.15
299.08	-31.35	14.37	-103.33
299.10	-55.79	14.64	-103.50
299.12	-46.60	14.87	-103.85
299.14	-52.54	15.08	-104.08
299.16	-38.94	15.25	-104.31
299.18	-61.63	15.56	-104.57
299.20	-49.10	15.69	-104.82
299.22	-65.19	15.84	-105.00
300.00	-72.66	15.96	-105.27
300.02	-51.17	16.05	-105.57
300.04	-56.72	16.22	-105.79
300.06	-39.80	16.39	-106.06
300.08	-61.42	16.62	-106.26
300.10	-92.39	16.74	-106.49
300.12	-54.65	16.73	-106.74
300.14	-64.23	16.92	-107.02
300.16	-120.65	17.04	-107.28
300.18	-126.73	16.87	-107.58
300.20	-78.21	16.77	-107.72
300.22	-43.73	16.78	-107.77
301.01	-12.44	16.91	-107.90
301.02	-35.01	17.04	-107.93
301.04	-120.18	17.19	-108.04
301.06	-20.37	17.14	-108.13
301.08	26.79	17.32	-108.20
301.10	-7.74	17.49	-108.11
301.12	.00	17.84	-108.16
301.14	-130.06	17.95	-108.16
301.16	81.05	17.83	-108.31
301.20	82.52	17.86	-108.11
301.22	136.41	17.87	-108.03
302.00	-96.66	17.77	-107.93
302.02	43.59	17.76	-108.02
302.04	10.76	17.82	-107.96
302.06	88.49	18.17	-107.89
302.08	83.33	18.18	-107.49
302.10	111.85	18.21	-107.22
302.12	108.38	18.13	-107.01
302.14	105.19	18.07	-106.82
302.16	77.52	17.99	-106.51
302.18	82.32	18.03	-106.32
302.20	87.69	18.08	-105.93
302.22	92.08	18.09	-105.67
303.00	108.37	18.08	-105.38
303.02	96.97	17.96	-105.00
303.04	102.26	17.91	-104.57
303.06	102.26	17.85	-104.28

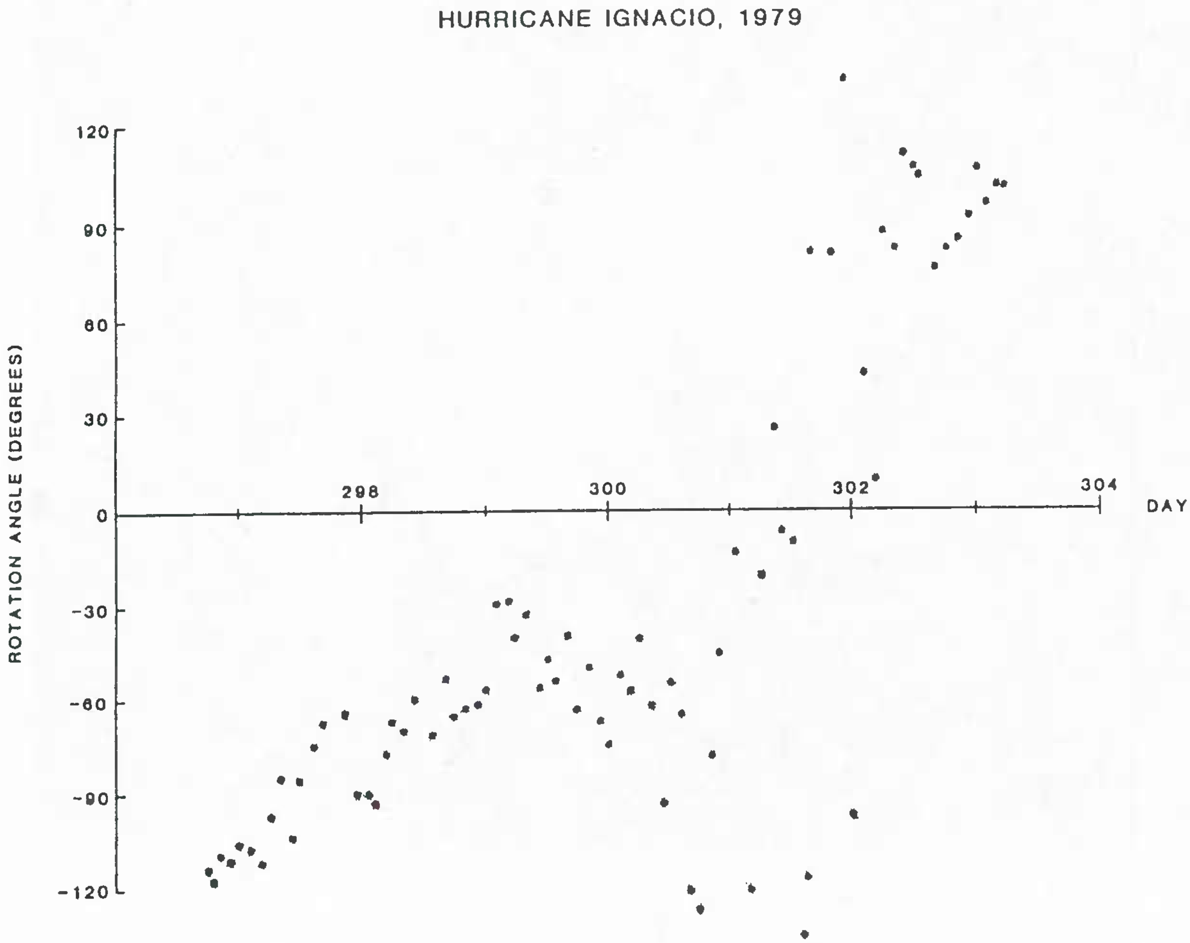


FIGURE 3-2: LAGRANGIAN ROTATION ANGLES FOR HURRICANE IGNACIO

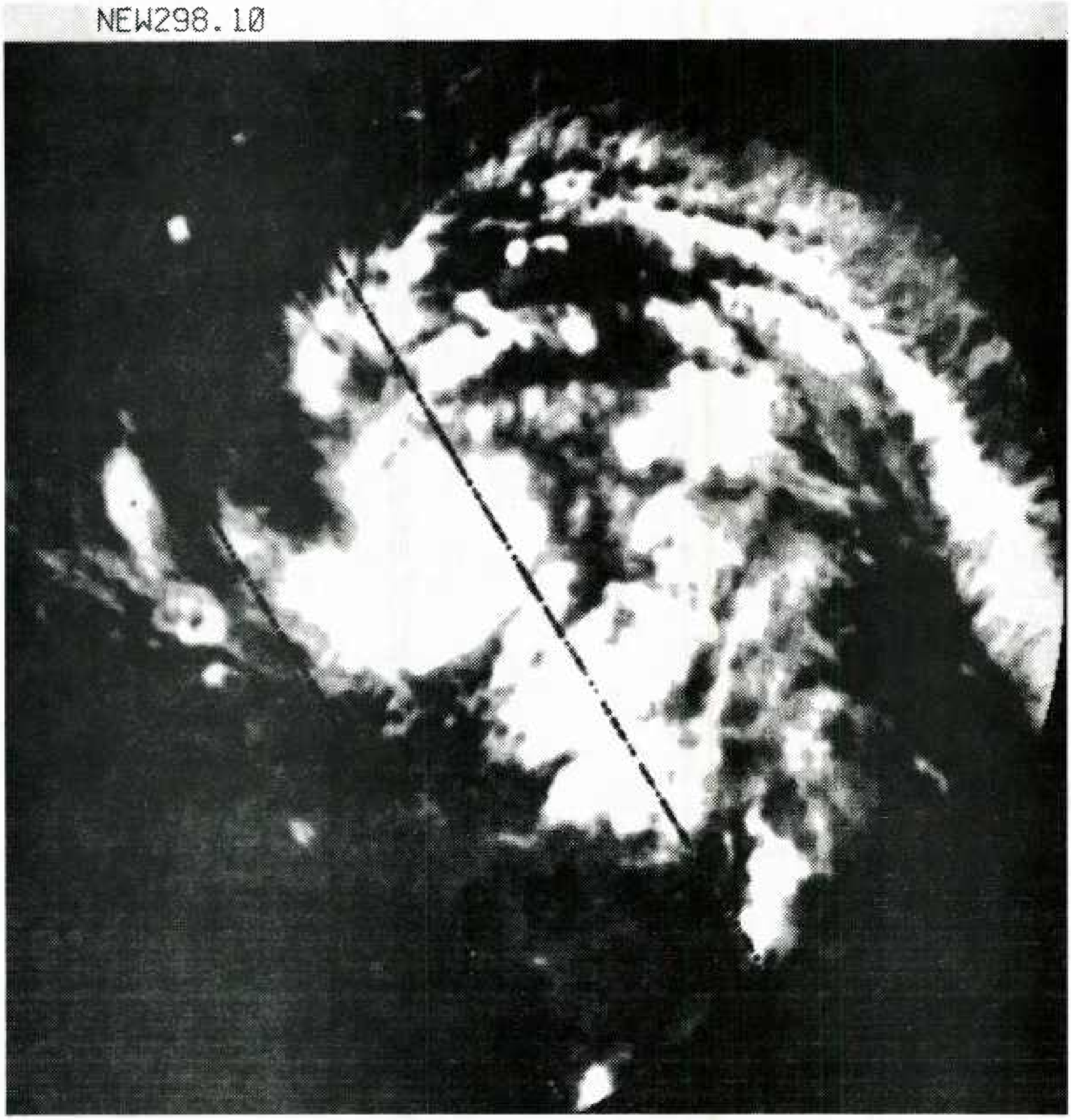


FIGURE 3-3

The image produced by TCROTAT from IGNACIO 298.10. From Table 3-1 we see that the rotation angle for this image was -59.04°.

FIGURE 3-4

The image produced by ITRPIMG by interpolating between the "before" and "after" images shown in Figures 3-3 and 3-5.

FIGURE 3-5

The image produced by TCROTAT from IGNACIO 298.14. From Table 3-1 we see that the rotation angle for this image was -70.61°.

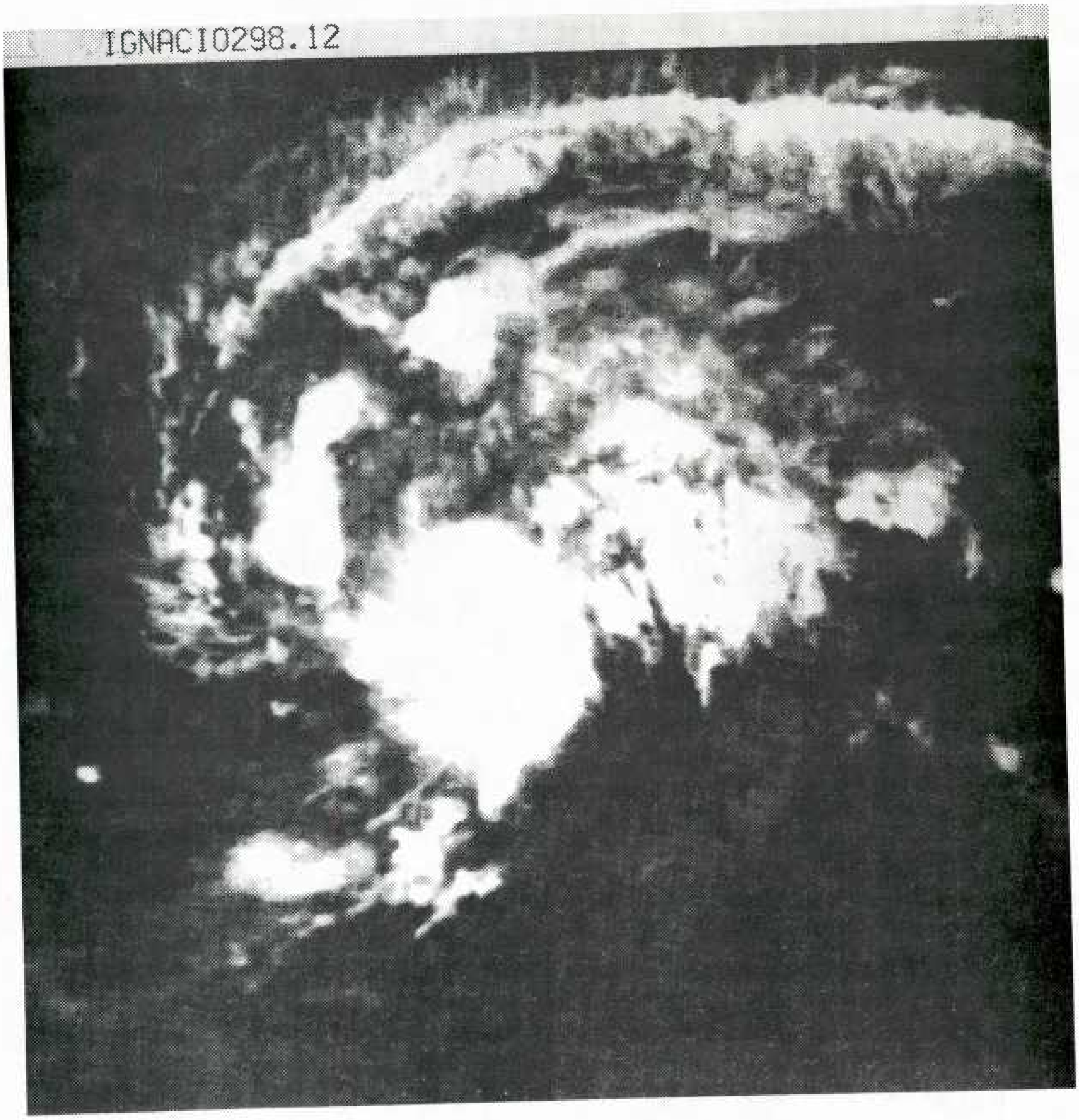


FIGURE 3-6

Image IGNACIO 298.12; the original image has been rotated during photocopy to give the same orientation as Figure 3-4, with which it should be compared.

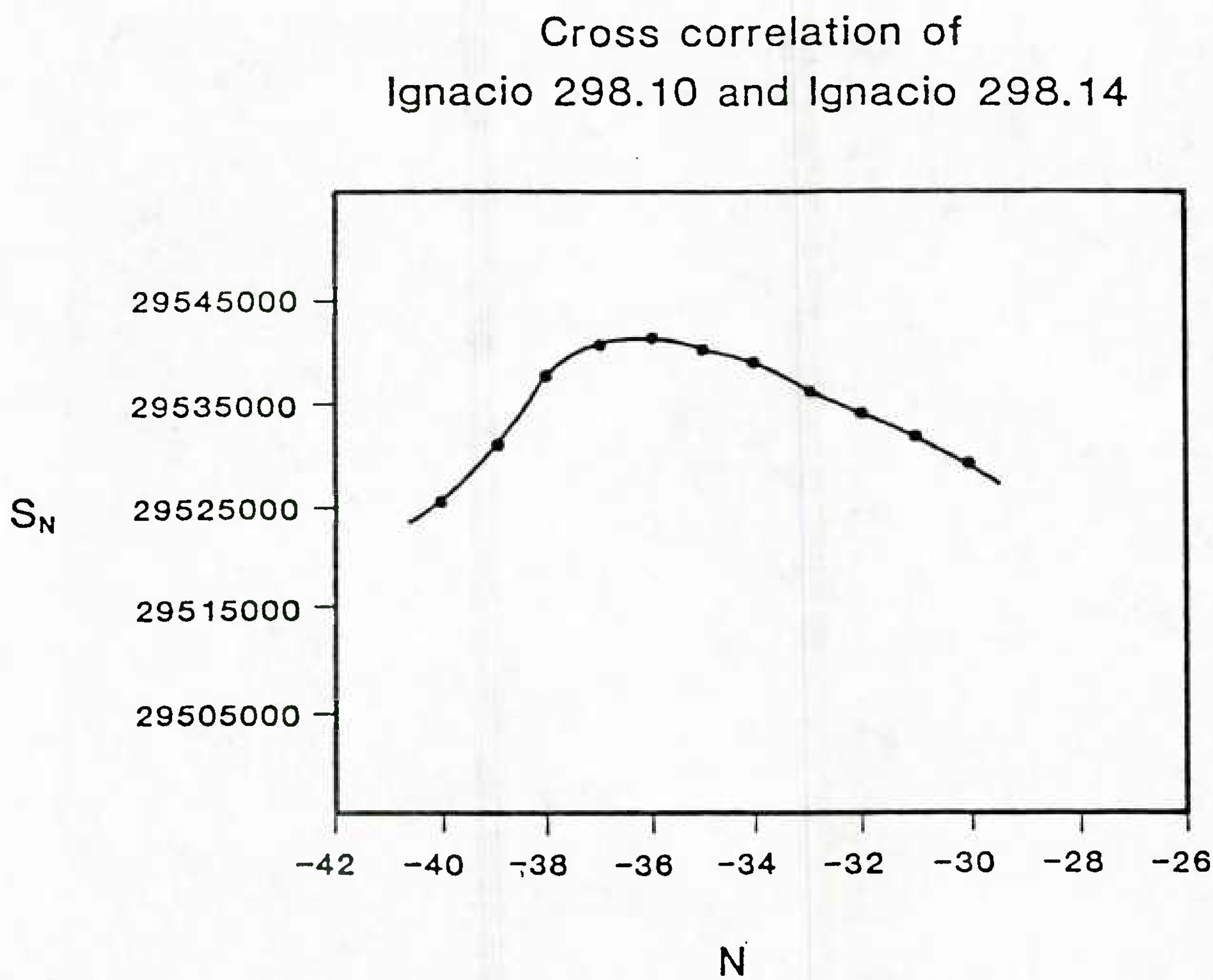


FIGURE 3-7: THE CROSS-CORRELATION FUNCTION, S_n, AS A FUNCTION OF THE NUMBER OF ANGULAR STEPS, N, FOR THE IMAGES SHOWN IN FIGURES 3-3 AND 3-5.

The radius chosen for this calculation was 200 pixels. The peak occurs at N=-36, which corresponds to an angle of -10.32°; each image is thus rotated half this amount, or -5.16°, prior to pixel-by-pixel interpelation.

3.3 Recommendations for Future Work

As interesting as the two sequences on videotape are, they do have a number of defects that should be corrected in future work. Below we list the most obvious shortcomings and suggest ways to overcome them. These defects were not corrected in the present contract because of time limitations. The biggest time problem is that of processing the images on SPC; usually not more than two or three images could be rotated in one day.

3.3.1 Center Locating

Errors in the placement of the storm centers can lead to serious errors in the calculated rotation angles for the images, especially when the storm is moving slowly. This can be seen in Figure 3-2; during days 300 and 301, especially, the calculated angles showed a lot of scatter. Consequently, many of the images in the second sequence on videotape are incorrect and actually confuse, rather than help, the study of the hurricane.

Another problem caused by inaccurate centering is in the interpolation process to replace missing images. The cross-correlation technique is based on the assumption that consecutive images in the Lagrangian frame will differ more in orientation than in overall structure. If the centers are misplaced on these images, however, then the cross-correleation will probably not give an accurate value of the relative orientation; hence, the interpolated image will not be as good as it should be.

To determine accurate storm centers we recommend the following interative procedure.

1. Use program TCIF to establish the first-guess centers as was done for this project.

2. Calculate both the velocity of the storm and the Lagrangian rotation angle for each image. This would require modifying program ANGLEL to include the velocity calculation.

3. Plot time sequences of both angle and velocity and then smooth. Hand smoothing would probably be accurate enough for these purposes, but if not a weighted 5 point or 7 point running average could be used.

4. Use the smoothed values of angle and velocity to recompute the storm track, then calculating new values of latitude/longitude of the storm center for each image. Program TCIF could then be run again to see if these values are reasonable; if not, adjustments can be made to establish a better center location.

5. Repeat steps 2 through 4 as many times as necessary to obtain a self-consistent set of angles and velocities that are acceptable to the user of program TCIF. Special attention should be paid to slow movement periods and to turning periods. A consistent rotation angle variation must be followed.

With the rotation angles obtained this way the Lagrangian sequence of images would be more meaningful.

3.3.2 Additional Screen Information

When viewing the image sequences on videotape, it would be very useful to have additional information on the screen. This would include:

1. The day and hour of the image;
2. The velocity of the storm;
3. The Lagrangian rotation angle;
4. The intensity (maximum wind velocity) of the cyclone, as obtained from other sources.

It would also be useful to run these sequences many times with different temperature slicing in order to emphasize different types of clouds and, in at least one sequence, to show the temperature structure of the underlying sea surface, where it is visible between the clouds.

This would require the development of new interactive software for the input of labels and the setting of video lockup tables.

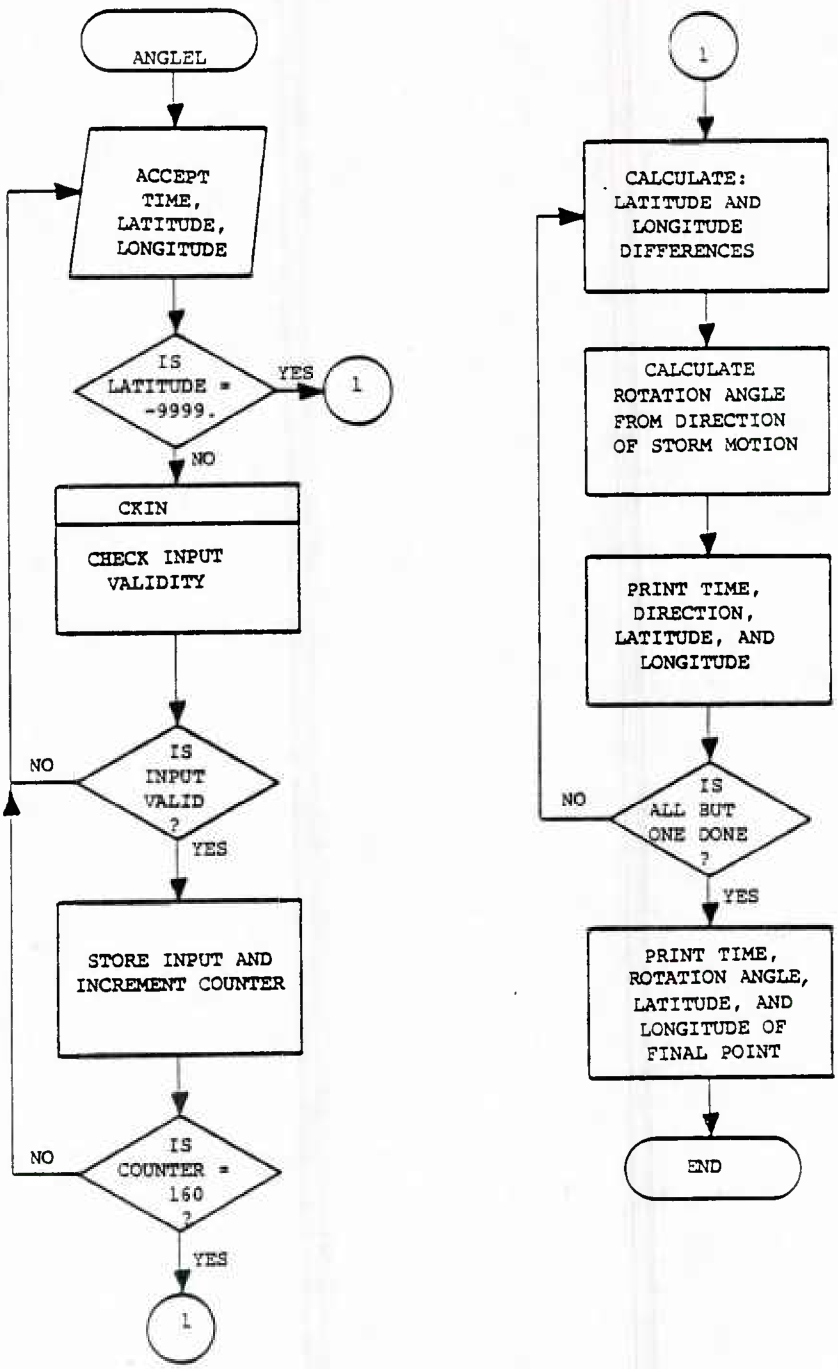
3.3.3 Mercator Projection

Hurricane Ignacio, as viewed from the GOES-West satellite, suffered some distortion in its images since it was located about 30° in longitude and 15° in latitude from the subsatellite point. Other hurricanes may suffer even more distortion. It would be useful to process the large sector image files with the program GOESMRC prior to processing with the programs discussed in this report so that geometrical distortions could be removed as well as possible. Fortunately, Tropical Cyclones are confined for the most part to tropical latitudes and so the distortion inherent in the mercator projection are not serious.

APPENDIX A

1. Flow chart for Program ANGLEL
2. Listing of ANGLEL
3. Flow chart for Subroutine CKIN
4. Listing of CKIN

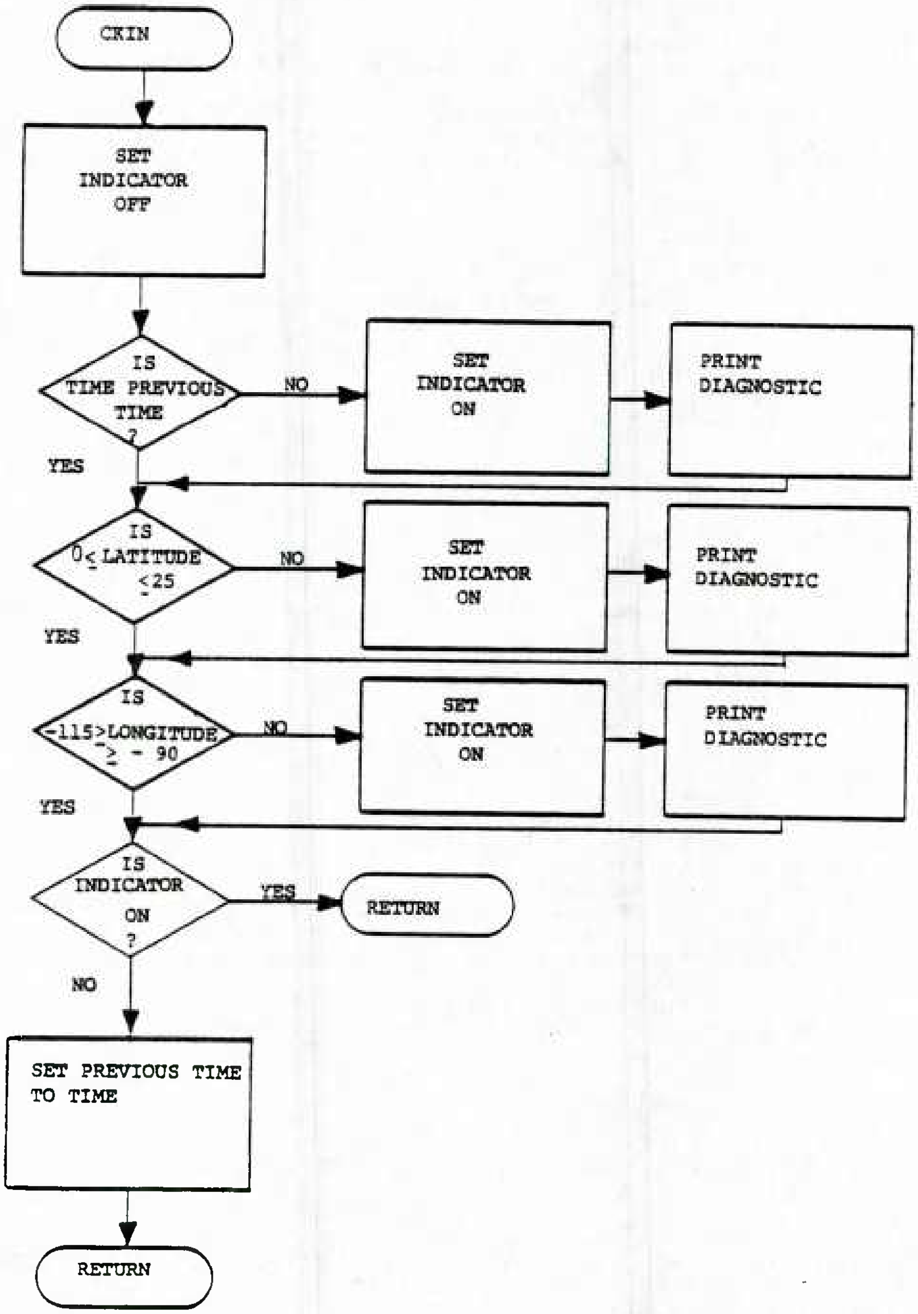
FLOW CHART
FOR PROGRAM ANGLEL



```
C       THIS PROGRAM CALCULATES THE ANGLE TO ROTATE A TROPICAL
C       CYCLONE IMAGE ACCORDING TO THE DIRECTION OF WHICH IT IS
C       MOVING.
C
C       INPUT:
C          XLAT,XLON     - MAX. 16Ø CENTER LOCATIONS OF CYCLONE
C                            (-9999. AFTER LAST POSITION)
C          TIME          -  TIME OF POSITION OF CYCLONE
C
C       OUTPUT
C                THETA   -  MAX 16Ø ANGLES
C                TIME    -  TIME OF POSITION OF T.CYCLONE
C
        DIMENSION XLAT(16Ø), XLON(16Ø), THETA(16Ø), TIME(16Ø)
        LOGICAL OK
        TYPE "PROGRAM ANGLEL"
        TYPE "THIS PROGRAM CALCULATES THE ANGLE TO ROTOTE"
        TYPE "A TROPICAL STORM."
        TYPE "ENTER TIME OF STORM (DAY.HR I.E. 299.Ø2),"
        TYPE "LATITUDE AND LONGITUDE POSITION AT THE PROMPT."
        TYPE "WHEN NO MORE DATA ENTER -9999. AT THE PROMPTS."
        LAST = Ø
        J = 1
        RADIAN = Ø.Ø17453
        DO 1Ø I=1,16Ø
                ACCEPT "ENTER TIME: ",TIME(J),"ENTER LATITUDE:",XLAT(J),
     $                "ENTER LONGITIUDE:",XLON(J),"<NL>"
                IF (XLAT(J).EQ.-9999.) GOTO 2Ø
                CALL CKIN(TIME(J),XLAT(J),XLON(J),OK)
                IF (.NOT.OK) GOTO 1Ø
                J = J+1
                LAST = LAST + 1
1Ø      CONTINUE
C
C CALCULATE THETA USING SLOPE BETWEEN POINTS
C
2Ø      PRINT 2
        IF (LAST.EQ.16Ø .AND. XLAT(LAST).NE.-9999.) LL=159
        IF (LAST.LE.16Ø .AND. XLAT(LAST).EQ.-9999.) LL=LAST-2
        IF (LAST.LE.16Ø .AND. XLAT(LAST).NE.-9999.) LL=LAST-1
        DO 3Ø L=1,LL
           DLAT = XLAT(L+1) - XLAT(L)
           DLON = XLON(L+1) - XLON(L)
           IF(DLAT.NE.Ø.) GO TO 22
           IF(DLON.GT.Ø.) THETA(L) = 9Ø.
           IF(DLON.LT.Ø.) THETA(L) = -9Ø.
           GO TO 32
22         HL = XLAT(L+1)
           IF (XLAT(L).GT.XLAT(L+1)) HL=XLAT(L)
           A = COS(HL*RADIAN) * DLON/DLAT
           THETARAD = ATAN(A)
           THETA(L) = THETARAD/RADIAN
           IF (XLAT(L+1).GT.XLAT(L)) GOTO 32    ;  STORM MOVE TOWARD EQUATOR
           IF(THETA(L).GT.Ø.) GO TO 25
           THETA(L) = THETA(L) + 18Ø.
           GO TO 32
25         THETA(L) = THETA(L) - 18Ø.
32         PRINT 3, TIME(L), THETA(L), XLAT(L), XLON(L)
3Ø      CONTINUE

          LL = LL + 1
          THETA(LL) = THETA(LL-1)
          PRINT 3, TIME(LL), THETA(LL), XLAT(LL), XLON(LL)
          GOTO 1ØØ
C
C FORMATS
2         FORMAT("1  TIME",T14,"ANGLE",T22,"LATITUDE",T31,"LONGITUDE"/
     $          "   DAY HR ",T12,"TO ROTATE"/)
3         FORMAT(1X,F7.2,3F1Ø.2)
C
C ERROR EXIT
C
1ØØ       STOP
          END
```

FLOW CHART
FOR SUBROUTINE CKIN



```
      SUBROUTINE CKIN(TIME,XLAT,XLON,OK,FIRST)
C THIS SUBROUTINE CHECKS THE VALIDITY OF THE INPUT DATA.
C
C INPUT:
C    TIME - TIME OF POSITION OF STORM
C    XLAT, XLON - LATITUDE, LONGITUDE POSITION OF STORM
C
C
      STATIC LASTME/Ø.Ø/
      INTEGER TIME
      LOGICAL   OK
C
      OK = .TRUE.
      GOTO 2Ø
1Ø    IF (LASTME.LT.TIME) GOTO 2Ø
      OK = .FALSE.
      TYPE "TIME INVALID, MUST BE LATER THAN PREVIOUS, TRY AGAIN"
2Ø    IF (XLAT.GE.Ø. .AND. XLAT.LE.25.) GOTO 3Ø
      OK = .FALSE.
      TYPE "INVALID LATITUDE MUST BE Ø TO 25.,TRY AGAIN"
3Ø    IF (XLON.LE.-9Ø. .AND. XLON.GE.-115.) GOTO 4Ø
      OK = .FALSE.
      TYPE "LONGITUDE INVALID, MUST BE -9Ø. TO -115, TRY AGAIN"
4Ø    IF (.NOT.OK) RETURN
      LASTME = TIME
      RETURN
      END
```

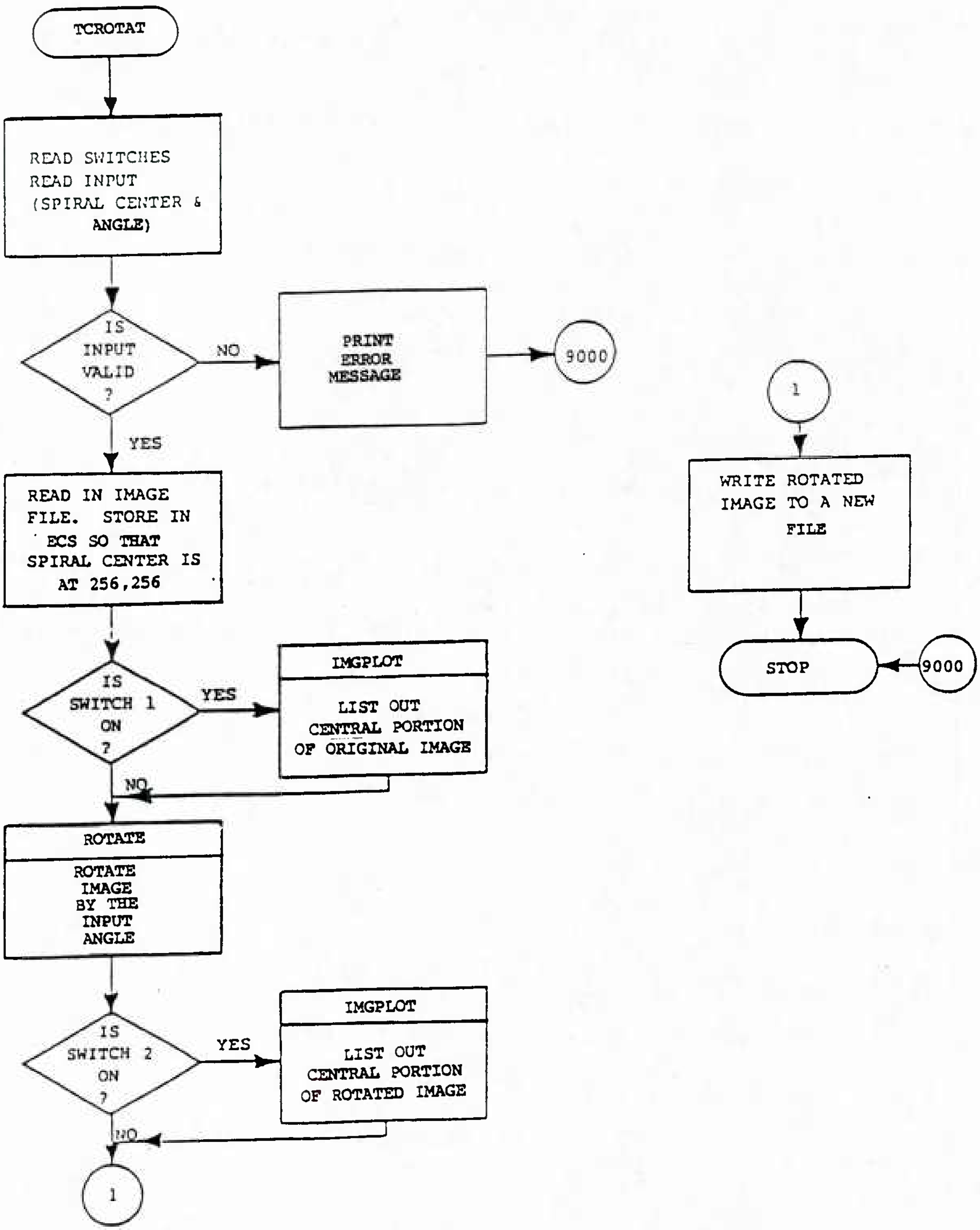
APPENDIX B

1. Listing of job control stream, JTCROTATE
2. Flow chart for Program TCROTAT
3. Listing of TCROTAT
4. Flow chart for Subroutine IMGPLOT
5. Listing of IMGPLOT
6. Flow chart for Subroutine ROTATE
7. Listing of ROTATE

JTCROTATE, ID=UL

```
VNULR,T2000,HD1,EC215,STSPC,S6.46      SUE UHRICH
RFE(0)
REWIND(OUTPUT)
ATTACH,ROTAT,TCROTATBIN,ID=UL.
LIBRARY(*FNWCLIB)
MAP(OFF)
REQUEST(IN,HD,L,NORING,VSN=71395)          VERIFY VSN
SKIPF(IN,11,17,B)                          VERIFY NUMBER OF FILES
COPYBF(IN,TAPE4)                           TO SKIP
UNLOAD(IN)
REWIND(ROTAT)
RFE(215)
ONSW(1)
ONSW(2)
ROTAT.
RFE(0)
REWIND(TAPE3)
REQUEST(OUT,HD,L,RING,VSN=71393)           VERIFY VSN
SKIPF(OUT,8,17,B)                          VERIFY NUMBER OF FILES
COPYBF(TAPE3,OUT)                          TO SKIP
RETURN(OUT)
*EOR
 222 269   -71.63                          INPUT CARD
*EOR                                        In this case:
*EOF                                          ICNTR = 222
                                              JCNTR = 269
                                              ANGLE = -71.63°
```

FLOW CHART
FOR PROGRAM TCROTAT

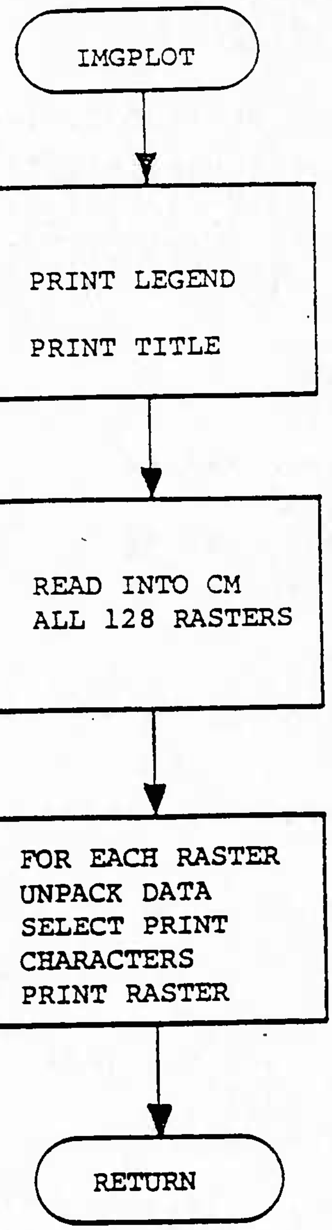


```
      PROGRAM TCROTAT(INPUT=200B,OUTPUT=200B,TAPE3=0,TAPE4=0)
C
C
C
C  THIS PROGRAM PROCESSES SATELITE IMAGES THAT ARE ON TAPE.
C  THE IMAGE IS READ IN AND CENTERED IN THE ↑WINDOW↑ ACCORDING TO
C  THE INPUT SPIRAL CENTER.  THE CENTERED IMAGE IS THEN ROTATED
C  ABOUT THIS CENTER.  LASTLY, THE CENTERED ROTATED IMAGE IS WRITTEN
C  BACK TO TAPE.
C
C  INPUT:
C     IMAGE    -  A SATELITE IMAGE ON FILE, TAPE4
C     ICNTR,JCNTR - PIXEL,RASTER LOCATION OF SPIRAL CENTER OF STORM.
C     ANGLE    -  ANGLE TO ROTATE IMAGE COUNTER-CLOCKWISE (DEGREES)
C
C  OUTPUT:
C     RIMAGE -   AN INTEGER FILE CONTAINING THE CENTERED IMAGE
C
C  PROCESSES:
C     READ IMAGE, ANGLE, AND CENTER LOCATION.
C     CENTERS IMAGE TO 256,256 PIXEL AND RASTER LOCATION
C     ROTATES IMAGE
C     PRINT ORIGINAL AND/OR IMAGE ON LINE-PRINTER
C
C  SWITCH SETTINGS:
C     SWITCH 1  ON  = PRINT OUT ORIGINAL IMAGE
C     SWITCH 1 OFF  = NO PRINT OUT
C     SWITCH 2  ON  = PRINT OUT ROTATED IMAGE
C     SWITCH 2 OFF  = DON↑T PRINT ROTATED IMAGE
C
C  INPUT FORMAT:
C     COLUMNS     PARAMETER
C      1 -  3     X-VALUE OF CENTER, LEFT-MOST POINT IS 1
C      5 -  7     Y-VALUE OF CENTER, TOP-MOST POINT IS 1
C      9 - 15     ANGLE TO ROTATE IMAGE (DEGREES, F7.2)
C
C  COMMON BLOCK - ECBLK
C     CONTAINS THE FIRST WORD ADDRESSES (FWA) OF EACH RECORD IN THE
C     IMAGE FILES, WHICH ARE STORED IN EXTENDED CORE STORAGE.
C         IEC1 - THE INPUT ORIGINAL IMAGE
C         IEC2 - THE ROTATED IMAGE DATA (NO HEADER OR DOCUMENATION)
C
      INTEGER BLOCK(100), HEADER(69)
      INTEGER IRASTER(512)
      COMMON /ECBLK/ IEC1(515), IEC2(512)
C
C----INITIALIZE AND READ SWITCH SETTINGS
C
      CALL SSWTCH(1,ISW1)
      CALL SSWTCH(2,ISW2)
C
C----READ INPUT AND CHECK VALIDITY
C
      READ 111,ICNTR,JCNTR,ANGLE
      IF (EOF(5LINPUT)) 9000,100
  111 FORMAT(I3,1X,I3,1X,F7.2)
  100 IF (ICNTR.GT.150 .AND. ICNTR.LT.362) GOTO 150
      PRINT 333,ICNTR,JCNTR,ANGLE
  333 FORMAT(* ERROR IN INPUT:*/
     1  5X,*CENTER= *,I3,*,*,I3,5X,*ANGLE=*,F7.2)
      GOTO 9000
  150 IF (JCNTR.GT.150 .AND. JCNTR.LT.362) GOTO 175
      PRINT 333,ICNTR,JCNTR,ANGLE
      GOTO 9000
  175 IF (ANGLE.LE.180.0 .AND. ANGLE.GE.-180.0) GOTO 200
      PRINT 333,ICNTR,JCNTR,ANGLE
  200 PRINT 444, ICNTR,JCNTR,ANGLE
  444 FORMAT(*1INPUT DATA CARD:*/
     2         *     CENTER LOCATED AT *,I3,*,*,I3/
     +         *     ANGLE TO ROTATE IS*,F7.2,* DEGREES*)
C
C
C
C  READ IMAGE FROM TAPE, CENTERING IMAGE AS READ IN.  THE IMAGE IS
C  515 RECORD BLOCKS LONG, AND EACH RECORD IS 256 16-BIT WORDS.
C  THE HEADER RECORD IS READ FIRST AND STORED LAST.  THE DATA IS ON
C  RECORDS 2 TO 513 AND IS STORED IN RASTER 1 TO 512.  THE DOCUMENATIC
C  RECORDS ARE ON 514 AND 515 AND ARE STORED AFTER THE DATA RECORDS.
```

```
C
      ID = 256-ICNTR
      JD = 256-JCNTR+1
      IBYT = IABS(ID)+1
      ILOC=0
C
C IF JD POSITIVE MOVE IMAGE DOWN, IF NEGATIVE MOVE UP
C IF ID POSITIVE MOVE IMAGE RIGHT, IF NEGATIVE MOVE LEFT
C FIRST ZERO OUT OF RANGE RASTERS
      CALL ZILCH(BLOCK,100)
      IF (JD.GT.1) GOTO 10
      IF (JD.EQ.1) GOTO 15
      IST = 512+JD
      DO 2 I=IST,512
         IEC1(I)=69*(I-1)
    2    CALL WRITEC(BLOCK,IEC1(I),69)
      GOTO 15
   10 IED = JD-1    $    ILOC=JD-1
      DO 12 I=1,IED
         IEC1(I)=69*(I-1)
   12    CALL WRITEC(BLOCK,IEC1(I),69)
C READ IN IMAGE
   15 REWIND 4
      BUFFER IN (4,1) (HEADER(1),HEADER(69))
      IF (UNIT(4)) 20,110,110
   20 IEC1(515)=69*514
      CALL WRITEC(HEADER,IEC1(515),69)
      DO 50 N=2,515
         CALL ZILCH(BLOCK,100)
         BUFFER IN (4,1) (BLOCK(1),BLOCK(69))
         IF (UNIT(4)) 25,110,110
   25    IF (N.GT.513) GOTO 30
         IF (JD.LT.1) GOTO 45
         ILOC = ILOC+1
         IF (ILOC.GE.513) GOTO 50
         IEC1(ILOC) = 69*(ILOC-1)
         IF (ID.EQ.0) GOTO 40
         IF (ID.LT.0) CALL BRPK(BLOCK,8,IBYT,IRASTER,8,1,512)
         IF (ID.GT.0) CALL BRPK(BLOCK,8,1,IRASTER,8,IBYT,512)
         CALL WRITEC(IRASTER,IEC1(ILOC),69)
         CALL ZILCH(IRASTER,100)
         GOTO 45
   30    IEC1(N-1) = 69*(N-2)
   40    CALL WRITEC(BLOCK,IEC1(N-1),69)
   45    JD = JD+1
   50 CONTINUE
      GOTO 900
C ERROR EXIT
  110 PRINT 777
      GOTO 10000
C
C FORMATS
C
  555 FORMAT(* ERROR --- PARITY, CONTINUING EXECUTION*)
  777 FORMAT(* ERROR --- EOF*)
C
C
C----PLOT IMAGE IF SWITCHES 1 AND/OR 2 IS ON, ROTATE IMAGE
C
  900 IF (ISW1.EQ.1) CALL IMGPLOT(1)
      CALL ROTATE(ANGLE),RETURNS(9000)
      IF (ISW2.EQ.1) CALL IMGPLOT(2)
C
C
C WRITE ROTATED IMAGE BACK TO TAPE. (IN FORMAT READIBLE BY D.G.)
C
C- - -WRITE HEADER BLOCK TO TAPE3
C
      N = 1
      CALL READEC(HEADER,IEC1(515),69)
      BUFFER OUT (3,1) (HEADER(1),HEADER(69))
      IF (UNIT(3)) 2020,2110,2110
C
C- - -WRITE DATA, 512 BLOCKS LONG.
 2020 DO 2030 N=2,513
         LOC = N-1
         CALL READEC(BLOCK,IEC2(LOC),69)
```

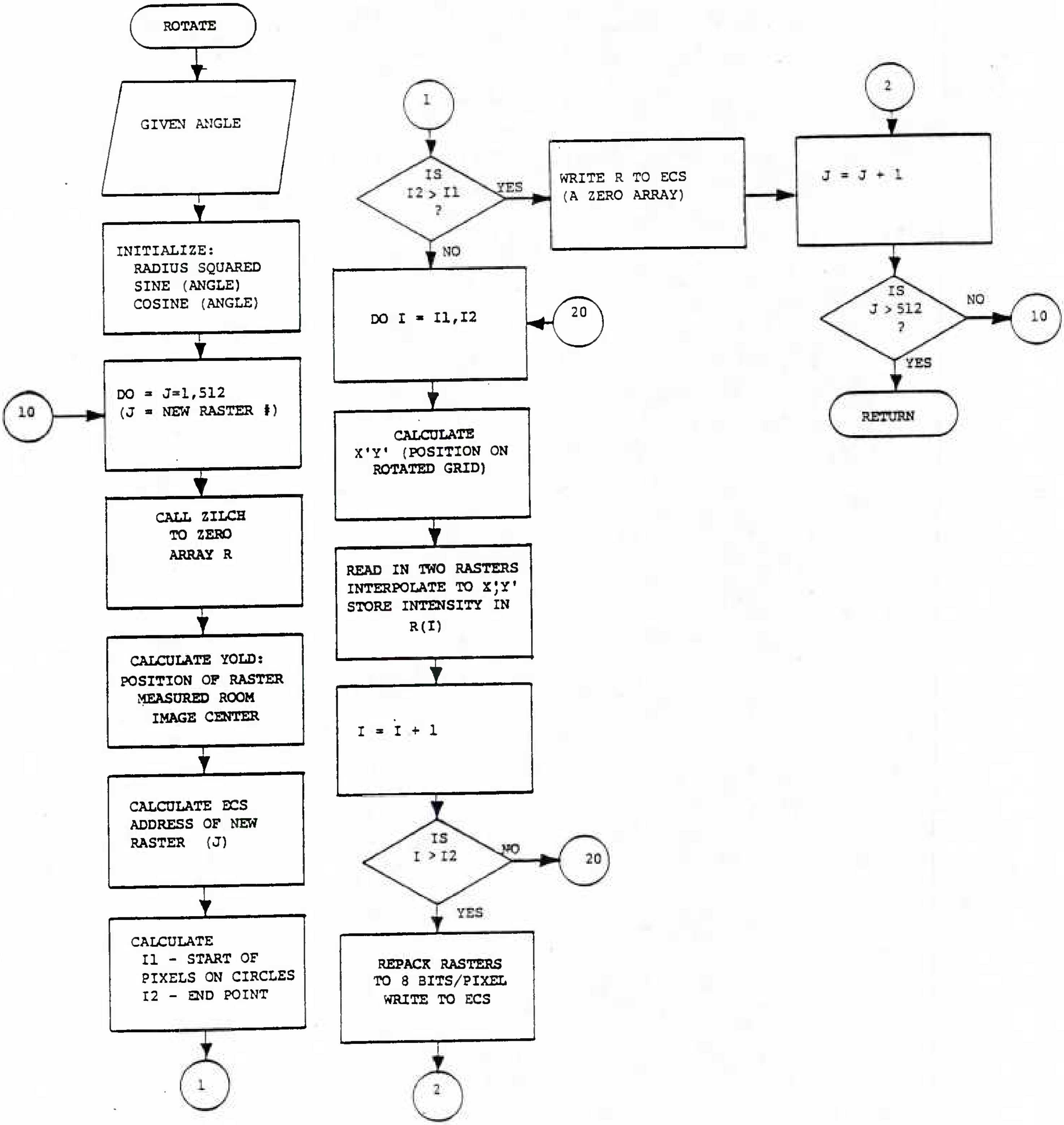
```
          BUFFER OUT (3,1) (BLOCK(1),BLOCK(69))
          IF (UNIT(3)) 2030,2110,2110
 2030 CONTINUE
C
C- - -WRITE DOCUMENTATION BLOCKS FROM ORIGINAL IMAGE
      DO 2040 N=514,515
          LOC = N-1
          CALL READEC(BLOCK,IEC1(LOC),69)
          BUFFER OUT (3,1) (BLOCK(1),BLOCK(69))
          IF (UNIT(3)) 2040,2110,2110
 2040 CONTINUE
      ENDFILE 3
      GOTO 10000
C
C- - -ERROR EXIT
C
 2110 PRINT 999,N
  999 FORMAT(* ERROR - PARITY OR EOF IN WRITING TAPE - NO. BLOCKS*,
     $           * READ =*,I5)
 9000 PRINT 9999
 9999 FORMAT(* ERROR, STOPPING EXECUTION*)
10000 CONTINUE
      STOP
      END
```

FLOW CHART
FOR SUBROUTINE IMGPLOT



```
      SUBROUTINE IMGPLOT(ITYP)
C
C THIS SUBROUTINE PLOTS THE IMAGE ON THE LINE-PRINTER
C
C INPUT:
C     IMAGE - AN INTEGER FILE CONTAINING THE INTENSITY VALUE OF
C             EACH PIXEL.
C     ECBLK - COMMON AREA WITH FWA OF IMAGE RASTERS
C             IEC1 - ORIGINAL IMAGE
C             IEC2 - ROTATED IMAGE
C      ITYP - 1 = ORIGINAL IMAGE
C             2 = ROTATED IMAGE
C
C OUTPUT:
C     A 128X128 LINE-PRINTER PLOT OF CENTRAL PORTION OF IMAGE
C
      DIMENSION IMAGE(128), ICHAR(16), ITYPE(2)
      INTEGER RASTER(128), BLOCK(8832)
      COMMON /ECBLK/ IEC1(515), IEC2(512)
      DATA ICHAR/1H ,1H.,1H:,1H-,1H+,1H*,1H1,1H2,
     $           1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H≡/
      DATA ITYPE/10H ORIGINAL ,10H ROTATED  /
C
C
C----PRINT TITLE AND LEGEND
C
      PRINT 111, ITYPE(ITYP)
  111 FORMAT(1H0,/
     1        59X,14H≠≠≠≠≠≠≠≠≠≠≠≠≠≠/
     2        59X,1H≠,12X,1H≠/
     3        59X,2H≠ ,A10,2H ≠/
     4        59X,14H≠ IMAGE PLOT ≠/
     5        59X,1H≠,12X,1H≠/
     6        59X,14H≠≠≠≠≠≠≠≠≠≠≠≠≠≠)
      PRINT 222
  222 FORMAT (1H0/
     1         53X,*LEGEND:*//
     2         53X,*INTENSITY     CHARACTER*/
     3         53X,*---------     ---------*/)
      INT1 = 0
      INT2 = 15
      DO 700 I=1,16
        PRINT 333, INT1, INT2, ICHAR(I)
        INT1 = INT1 + 16
        INT2 = INT2 + 16
  700 CONTINUE
  333 FORMAT(53X,I4,*-*,I4,8X,A1)
C
C----ITYP = 1, ORIGINAL IMAGE   ---   ITYP = 2, ROTATED IMAGE
C
      LW = -68
      PRINT 666, ITYPE(ITYP)
  666 FORMAT(1H1,*FILE OF*,A10,*LISTED*)
      IF (ITYP.EQ.1) IEC=IEC1(193)
      IF (ITYP.EQ.2) IEC=IEC2(193)
      CALL READEC(BLOCK,IEC,8832)
      DO 1100 J=1,128
         LW = LW +69
C----UNPACK 128 PIXELS IN A RASTER STARTING WITH THE 193RD PIXEL.
         CALL BRPK(BLOCK(LW),8,193,IMAGE,60,1,128)
         K = 0
         DO 1000 I=1,128
            K = K + 1
            INT = IMAGE(I)
            INT = INT/16 +1
            IF (INT.LT.1 .OR. INT.GT.16) PRINT 888,INT
            IF (INT.LT.1 .OR. INT.GT.16) INT = 16
            RASTER(K) = ICHAR(INT)
 1000    CONTINUE
         PRINT 777, (RASTER(L),L=1,128)
 1100 CONTINUE
  777 FORMAT(1H ,128(A1))
  888 FORMAT(* ERROR, INT=*,I5)
C
C----ALL DONE
C
      RETURN
```

FLOW CHART
FOR SUBROUTINE ROTATE



```
      SUBROUTINE ROTATE(ANGLE),RETURNS(XXX)
C
C THIS PROGRAM ROTATES THE IMAGE OF SIZE (M X N) ABOUT THE CENTER
C BY A GIVEN ANGLE (WHERE -180.1<ANGLE<180.1)
C
C INPUT:
C    IMAGE  = IMAGE ON MASS STORAGE FILE, TAPE1.
C    ANGLE  = THE ANGLE TO ROTATE THE IMAGE (COUNTERCLOCKWISE)
C    ECBLK  = COMMON BLOCK CONTAINING FWA OF EACH RASTER IN E.C.S.
C
C OUTPUT:
C    RIMAGE  = ROTATED IMAGE ON MASS STORAGE FILE, TAPE2.
C
C
      INTEGER IMAGE(1024), RIMAGE(69), RASTER(138)
      INTEGER R(512)
      COMMON /ECBLK/ IEC1(515), IEC2(512)
C
C----ROTATE AXIS ABOUT CENTER(255.,255.)
C
      IF (ANGLE.EQ.0.0) RETURN XXX
      RDRD = 255.0 * 255.0
      ARAD = ANGLE * 0.01745329
      SINANG = SIN(ARAD)
      COSANG = COS(ARAD)
      DO 200 J=1,512
         CALL ZILCH(R,512)
         IEC2(J)=69*(514+J)
         Y = J-1
         YOLD = 255.-Y
         X2X2 = RDPD - (YOLD)*(YOLD)
         IF (X2X2.LT.0.) GOTO 100
         X2 = SQRT(X2X2) + 255.
         I2 = X2 + 1.0
         I1 = 511.5 - X2
         IF (I2.GT.I1) GOTO 125
 100     CALL WRITEC(R,IEC2(J),69)
         GOTO 200
 125     DO 150 I=I1,I2
            XOLD = I-256
C
C----CALCULATE NEW POINT XPRIME, YPRIME
C
            XPRIME = (XOLD*COSANG) + (YOLD*SINANG)
            YPRIME = (YOLD*COSANG) - (XOLD*SINANG)
            IX = X = XPRIME + 255.
            IY = Y = 255. - YPRIME
            IF (IX.LT.1 .OR. IX.GE.512) GOTO 150
            IF (IY.LT.1 .OR. IY.GE.512) GOTO 150
C
C----INTERPOLATE INTENSITIES
            CALL READEC(RASTER,IEC1(IY),138)
            CALL BRPK(RASTER,8,1,IMAGE,60,1,512)
            CALL BRPK(RASTER(70),8,1,IMAGE(513),60,1,512)
            PO = IMAGE(IX)
            P1 = IMAGE(IX+1)
            P2 = IMAGE(IX+512)
            P3 = IMAGE(IX+513)
            DX = X - FLOAT(IX)
            DY = Y - FLOAT(IY)
            RINT = (P1-PO)*DX + (P2-PO)*DY +
     $             (P3+PO-P2-P1)*DX*DY + PO
            R(I)  = RINT+.5
 150     CONTINUE
         CALL BRPK(R,60,1,RIMAGE,8,1,512)
         CALL WRITEC(RIMAGE,IEC2(J),69)
 200  CONTINUE
      RETURN
      END
```

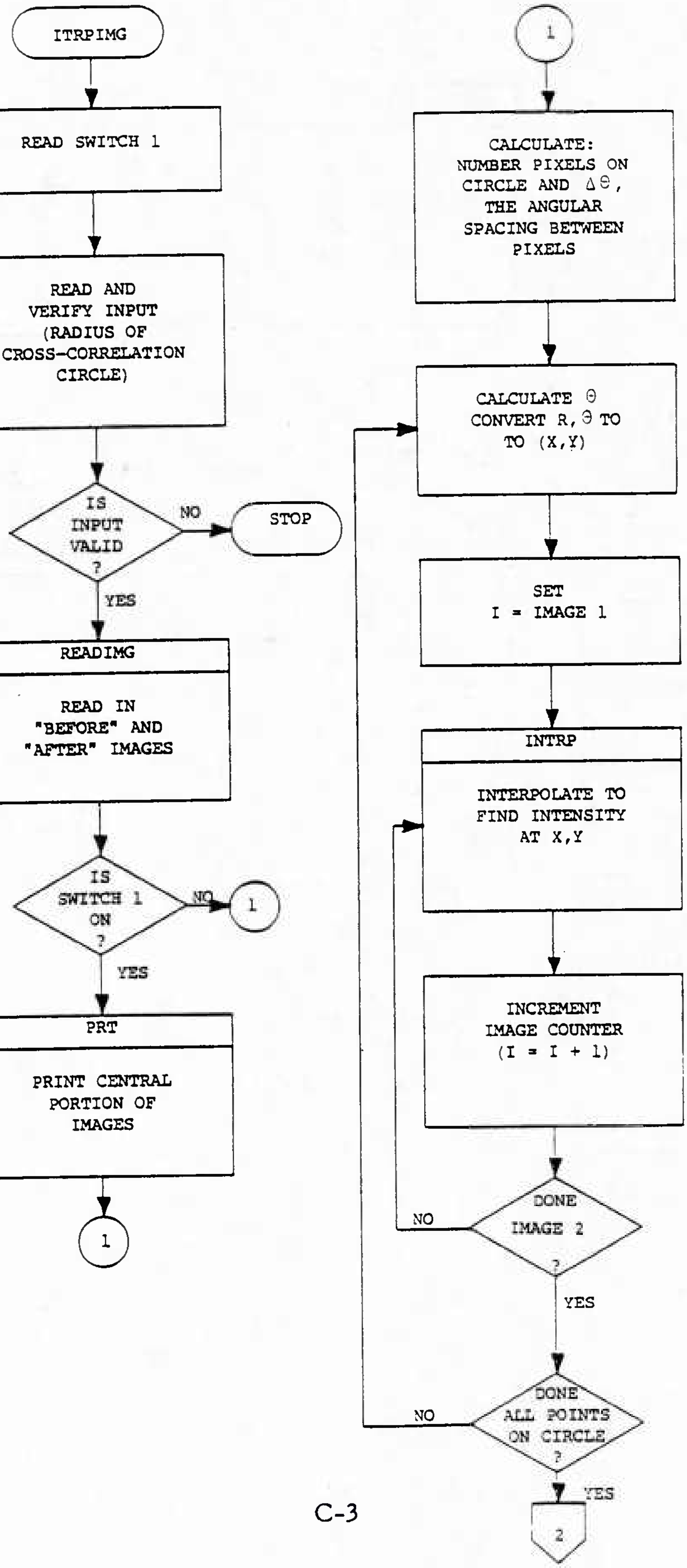
APPENDIX C

1. Listing of job control stream, JINTERPOLATE
2. Flow chart for Program ITRPIMG
3. Listing of ITRPIMG
4. Flow chart for Subroutine READIMG
5. Listing of READIMG
6. Flow chart for Subroutine SUMSN
7. Listing of SUMSN
8. Flow chart for Subroutine ROTAVG
9. Listing of ROTAVG
10. Flow chart for Subroutine INTRP
11. Listing of INTRP
12. Flow chart for Subroutine PRT
13. Listing of PRT
14. Flow chart for Subroutine RITIMG
15. Listing of RITIMG

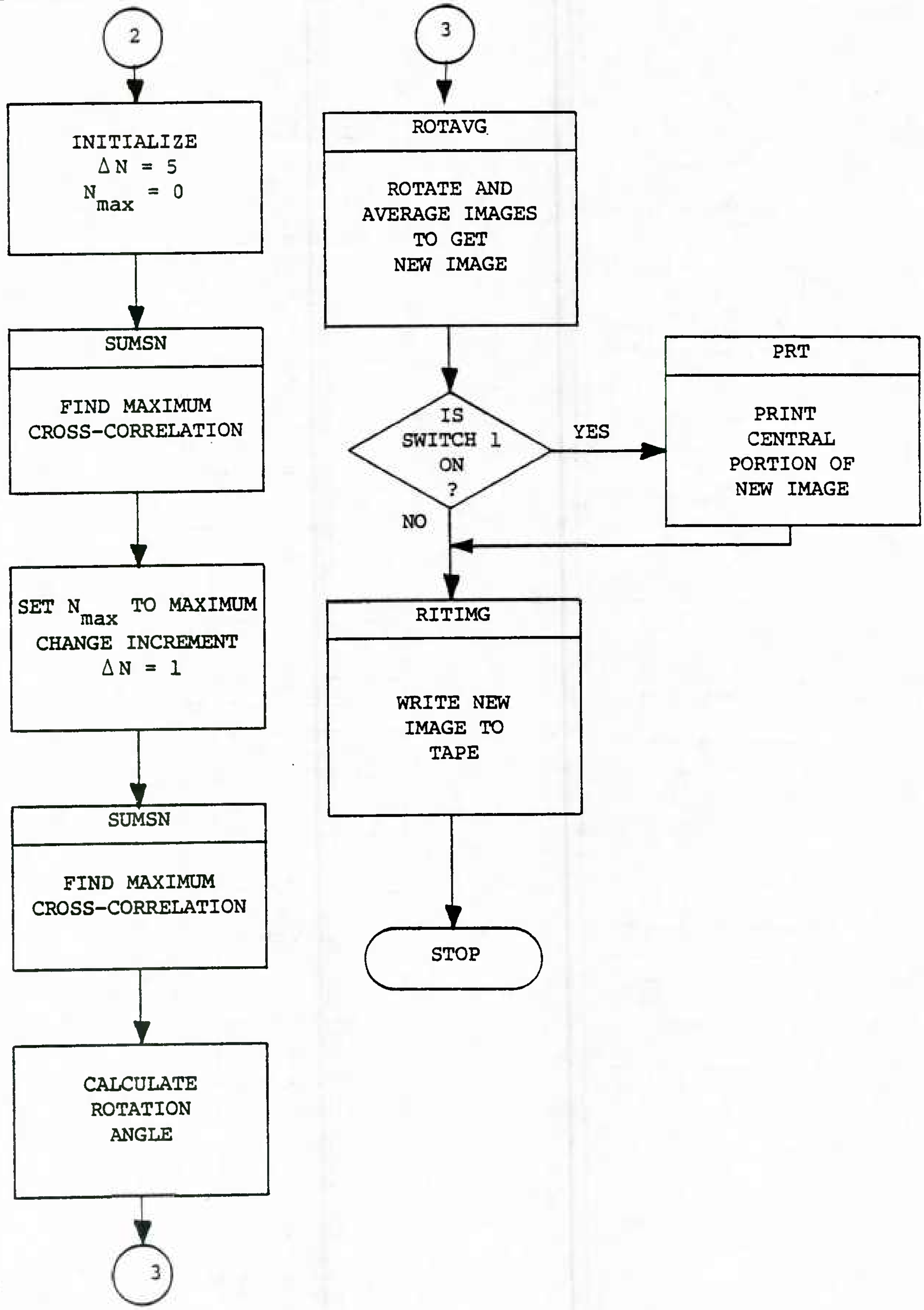
JINTERPOLATE, ID = UL

```
VNULI,T4000,EC320,HD1,,STSPC,S6.46
RFE,0.
REWIND,OUTPUT
REWIND,OUTPUT.
ATTACH,ITRP,INTERPOLATEBIN,ID=UL.
LIBRARY,*FNWCLIB.
MAP,OFF.
REQUEST,IN,VSN=71396,HD,L,NORING.<——— VERIFY VSN
SKIPF,IN,18,17,B.<———————————————————— VERIFY NUMBER OF FILES TO SKIP
COPYBF,IN,TAPE3.
COPYBF,IN,TAPE4.
REWIND,TAPE3,TAPE4.
UNLOAD,IN.
ONWS,1.<—————————————————————————————— PRINT
RFE,320.
ITRP.
RFE,0.
RETURN,ITRP.
REWIND,TAPE5.
REQUEST,OUT,VSN=71397,HD,L,RING.<————— VERIFY VSN
SKIPF,OUT,2,17,B.<———————————————————— VERIFY NUMBER OF FILES TO SKIP
COPYBF,TAPE5,OUT.
RETURN,OUT.
EXIT,S.
*EOR
150<————————————————————————————————— INPUT CARD; RADIUS OF CROSS-
                                                  CORRELATION CIRCLE
*EOR
*EOF
```

FLOW CHART
FOR PROGRAM ITRPIMG



ITRPIMG

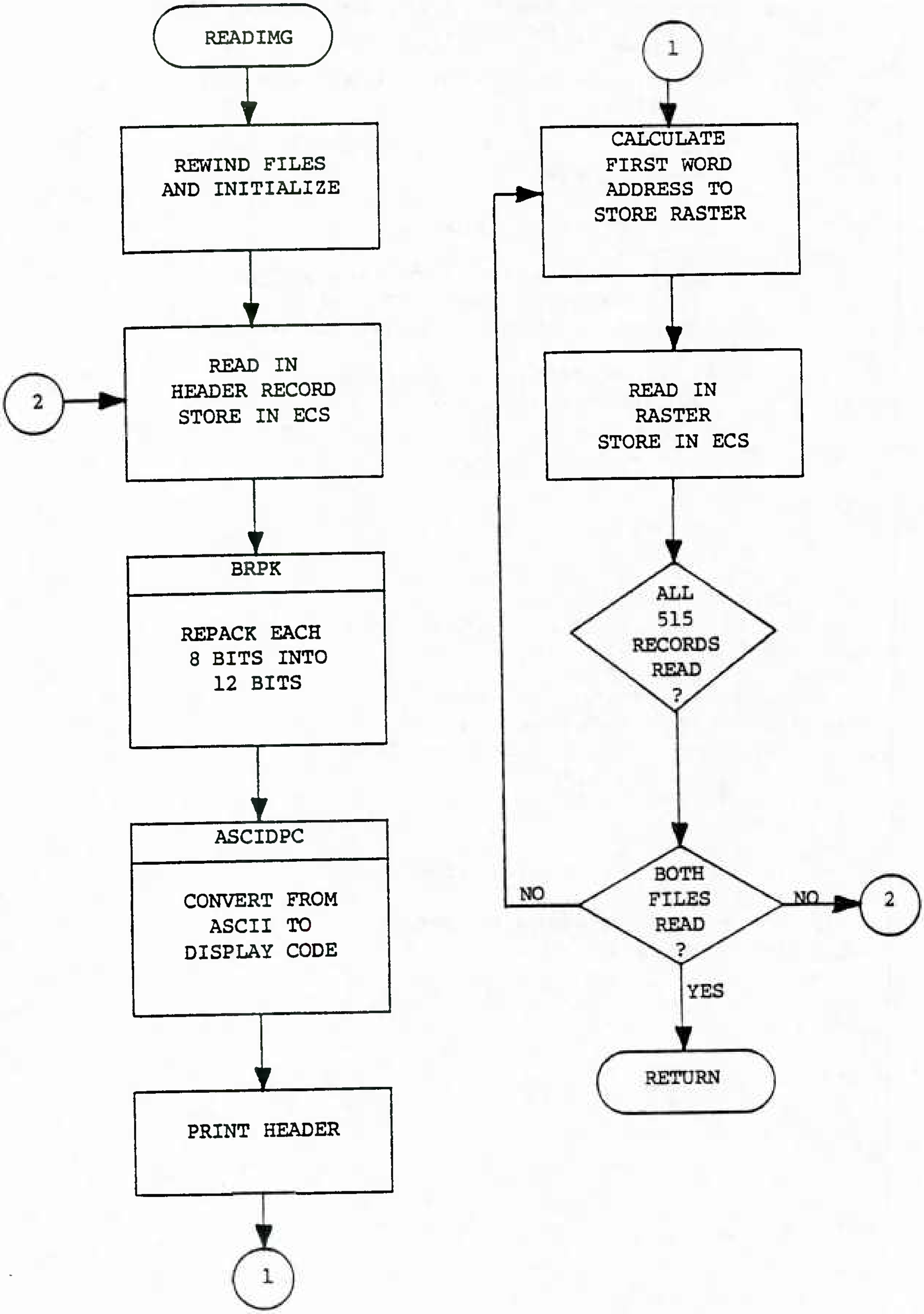


2
INITIALIZE
ΔN = 5
N_{max} = 0
SUMSN
FIND MAXIMUM CROSS-CORRELATION
SET N_{max} TO MAXIMUM
CHANGE INCREMENT
ΔN = 1
SUMSN
FIND MAXIMUM CROSS-CORRELATION
CALCULATE ROTATION ANGLE
3
3
ROTAVG
ROTATE AND AVERAGE IMAGES TO GET NEW IMAGE
IS SWITCH 1 ON ?
YES
NO
PRT
PRINT CENTRAL PORTION OF NEW IMAGE
RITIMG
WRITE NEW IMAGE TO TAPE
STOP

```
      PROGRAM ITRPIMG(INPUT=200B,OUTPUT=200B,TAPE3=0,TAPE4=0,TAPE5=0)
C
C THIS PROGRAM CREATES AN IMAGE, WHICH IS INTERPOLATED FROM TWO
C SATILLITE IMAGES READ FROM TAPE.  FROM A RADIUS OF AN ANNULUS
C READ IN, CROSS CORRELATION IS PERFORMED TO DETERMINE THE OPTIMUM
C ANGLE TO ROTATE THE IMAGES INPUT TO CREATE THE NEW IMAGE.
C
C INPUT WILL BE TWO IMAGES, AND THE RADIUS SPECIFIED ON THE INPUT CARD.
C
C OUTPUT: WILL BE THE INTERPOLATED IMAGE AND ANY CALCULATIONS MADE
C         DURING THE RUN OF THIS PROGRAM.
C
C SWITCH 1 ON: PRINT OUT BEFORE AND AFTER IMAGE, AND ALSO
C              PRINT CENTRAL PORTION OF INTERPOLATED IMAGE (128X128).
C
C
      COMMON /ECBLK/  IEC1(515), IEC2(515), IEC3(512)
      COMMON /INTENS/ IMG1(1602), IMG2(1602)
C
      CALL SSWTCH(1,IS1)
      PRINT 1
    1 FORMAT(1H0,T25,29H*     PROGRAM INTERPOLATE     *//
     $     1X,T11,50HTHIS PROGRAM WILL INTERPOLATE AN IMAGE BETWEEN TWO
     $        ,8H OTHERS./)
C
C----READ INPUT AND VERIFY
      READ 2,IRADIUS
    2 FORMAT(I3)
      PRINT 3, IRADIUS
    3 FORMAT(1H0,*RADIUS EQUALS *,I3)
      IF (IRADIUS.GE.1 .AND. IRADIUS.LE.255) GOTO 25
      PRINT 4
    4 FORMAT(1H0,50HRADIUS MUST BE GREATER THAN ZERO AND LESS THAN 255/
     +        1H ,21HPROGRAM STOPPING NOW.)
      GOTO 90
C
C- - -READ IN IMAGES AND PRINT IF SWITCH 1 IS ON
C
   25 CALL READIMG,RETURNS(90)
      IF (IS1.EQ.2) GOTO 30
      CALL PRT(1)
      CALL PRT(2)
C
C CONVERT X,Y LOCATIONS ON CIRCLE TO R,THETA COORDINATES.
C CORRESPONDING INTENSITIES ARE STORED IN THE COMMON BLOCK, INTENS.
C
   30 RAD = IRADIUS
      DTHETA = 1./RAD
      NOPTS = 2. * 3.14159265 * RAD
      PRINT 5, NOPTS
    5 FORMAT(*0NUMBER OF POINTS ON CIRCLE TO CONVERT =*,I5)
      DO 40 I=1,NOPTS
        THETA = (I-1)*DTHETA
        X = RAD * COS(THETA) + 255.0
        Y = 255.0 - (RAD * SIN(THETA))
```

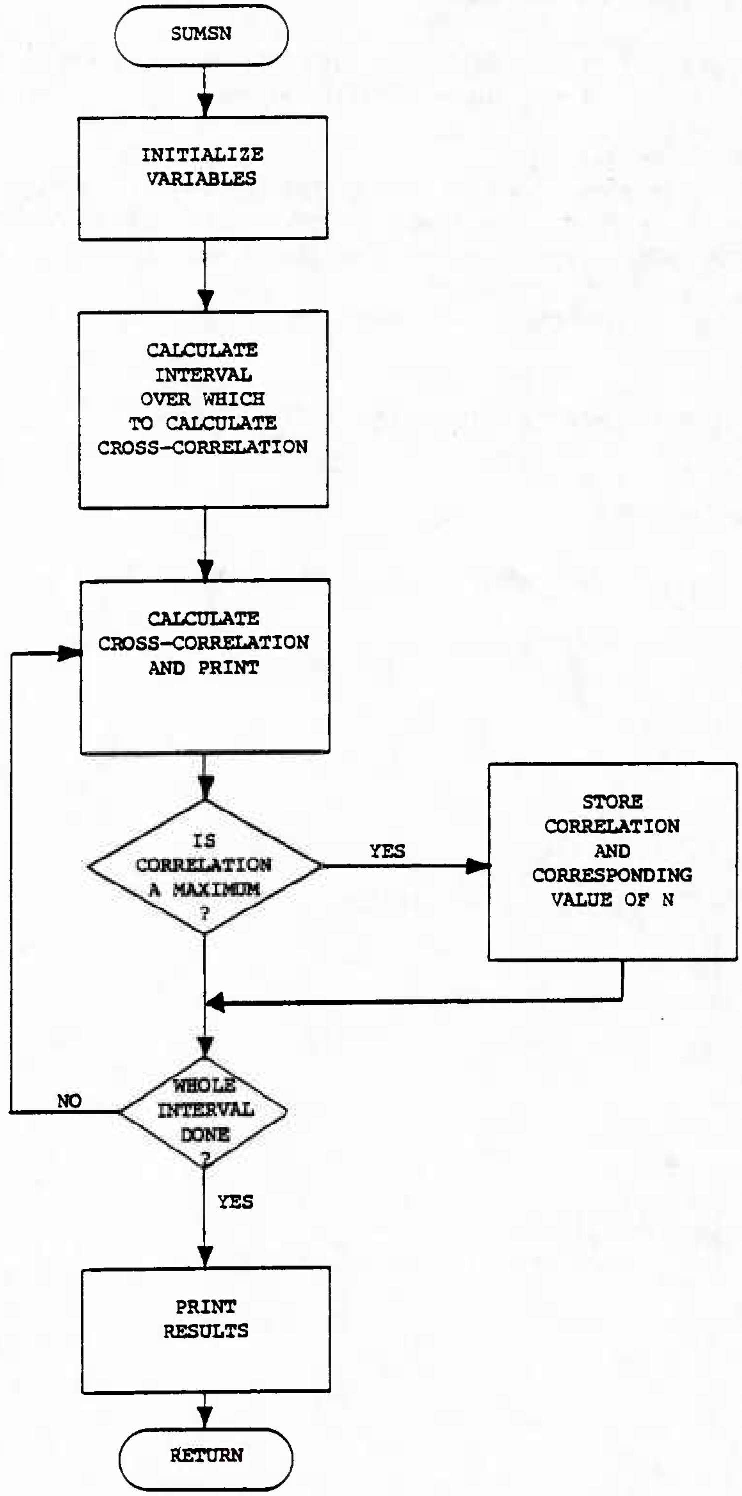
```
C- - -STORE INTENSITY IN ARRAY BY I EQUAL THETA STEP SIZE
C- - -IMAGE 1
         CALL INTRP(1,X,Y,INT)
         IMG1(I)=INT
C- - -IMAGE 2
         CALL INTRP(2,X,Y,INT)
         IMG2(I)=INT
   40 CONTINUE
C
C CALCULATE CROSS-CORRELATION. DONE IN TWO STEPS.
C      NINC = INCREMENT OF STEP SIZE
C      NMAX = VALUE OF N, AT MAX CORRELATION
C
       NMAX = 0
       NINC = 5
       CALL SUMSN(NOPTS,NINC,NMAX)
       NINC = 1
       CALL SUMSN(NOPTS,NINC,NMAX)
C
C----COMPUTE ANGLE AND ROTATE IMAGES
       ANGLE = NMAX * DTHETA/2.
       A = ANGLE/.01745329
       PRINT 8, ANGLE,A
     8 FORMAT(*0NOW ROTATING IMAGE A BY *,F7.4,* RADIANS OR *,F7.2,
      $        * DEGREES*)
       IF (ANGLE.EQ.0.) GOTO 90
       CALL ROTAVG(ANGLE)
       IF (IS1.EQ.1) CALL PRT(3)
C WRITE NEW IMAGE TO TAPE
       CALL RITIMG
90     CONTINUE
  100 STOP
       END
```

FLOW CHART
FOR SUBROUTINE READIMG



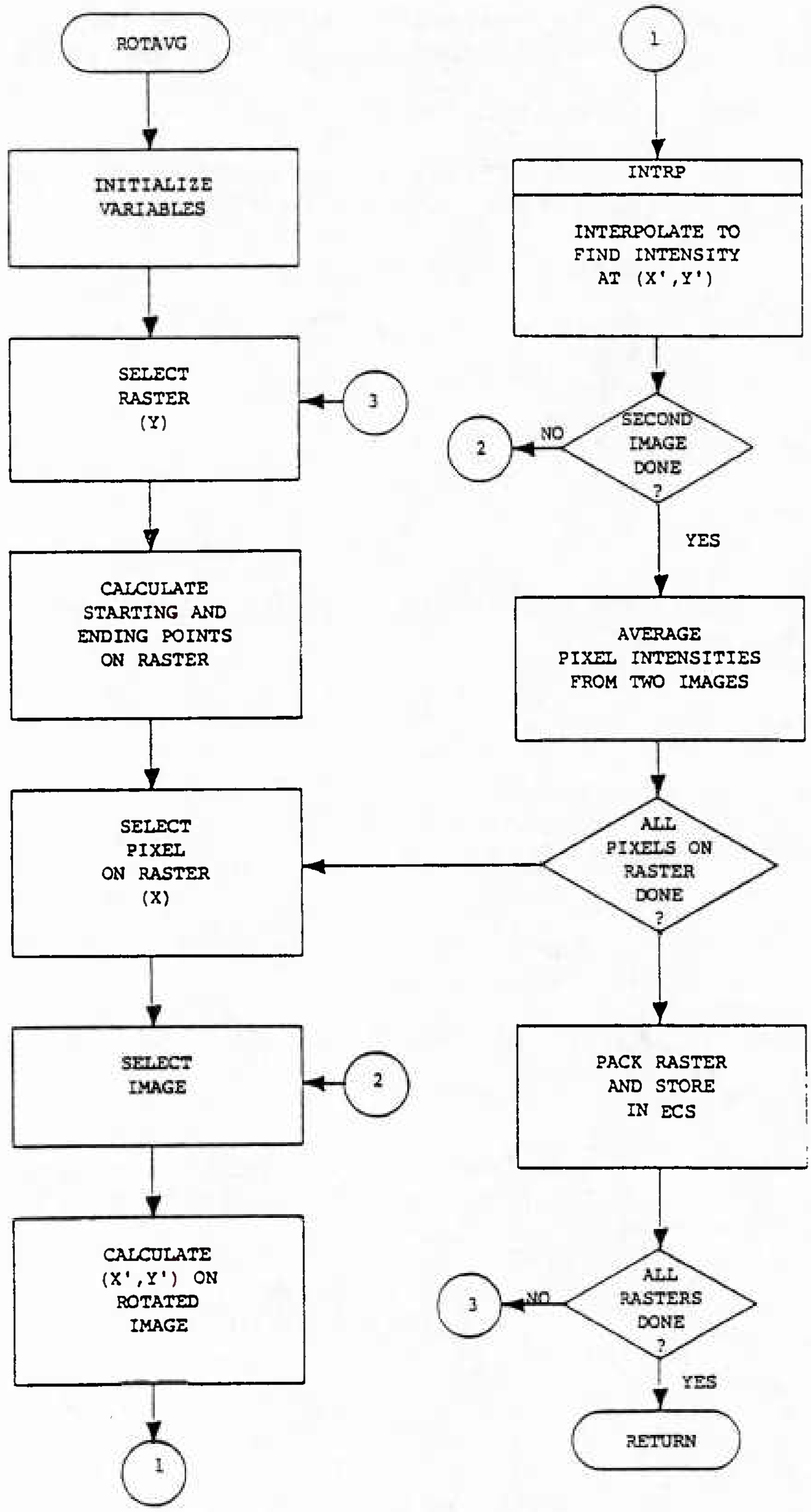
```
      SUBROUTINE READING,RETURNS(XXX)
C
C THIS SUBROUTINE READS IN TWO IMAGE FILES. THE BEFORE IMAGE IS ON
C TAPE3 AND THE AFTER IMAGE IS ON TAPE4.
C
C FORMAT OF EACH IMAGE FILE (RECORD = 256 16-BIT WORDS):
C    RECORD NO.       CONTENTS
C      1              HEADER
C      2-513          DATA
C    514-515          DOCUMENATION
C
C FORMAT OF OUTPUT FILES (RECORD=69 WORDS, ):
C    RECORD NO.       CONTENTS
C      1-512          DATA (512 8-BIT PIXELS PER RECORD)
C    513-514          DOCUMENTATION (UNCHANGED)
C    515              HEADER (UNCHANGED RECORD FORMAT)
C
      DIMENSION IMAGE(69),IHEADER(160),IWK(160)
      COMMON /ECBLK/ IEC(1030), IEC3(512)
C
C- - -READ TWO IMAGE FILES
C
      REWIND 4
      REWIND 3
      NW=1    $   IFILE=3
    5 N=1
C STORE HEADER RECORD LAST
      BUFFER IN (IFILE,1) (IMAGE(1),IMAGE(69))
      IF (UNIT(IFILE)) 10,110,110
   10 IEC(NW+514) = 69*(NW+513)
      CALL WRITEC(IMAGE,IEC(NW+514),69)
      CALL BRPK(IMAGE,8,1,IHEADER,12,1,80)
      CALL ASCIDPC(IHEADER,16,IWK,16,LDPC,IERR)
      IF (IERR.EQ.1) PRINT 111,LDPC
      PRINT 222,(IWK(L),L=1,LDPC)
      DO 100 N=2,515
        IEC(NW)=69*(NW-1)
        BUFFER IN (IFILE,1) (IMAGE(1),IMAGE(69))
        IF (UNIT(IFILE)) 50,110,110
C- - -STORE DATA AND DOCUMENTATION RECORDS
   50   CALL WRITEC(IMAGE,IEC(NW),69)
        NW=NW+1
  100 CONTINUE
      IFILE=IFILE+1
      NW=NW+1
      IF (IFILE.EQ.4) GOTO 5
      RETURN
C
C- - -ERROR EXIT
C
  110 PRINT 333, IFILE,N
      RETURN XXX
C
C- - -FORMATS
C
  111 FORMAT(* ERROR IN ASCII TO DISPLAY CODE CONVERSION, LDPC =*,I6)
  222 FORMAT(* IMAGE FILE HEADER IS *,8A10)
  333 FORMAT(* ERROR -- PARITY OR EOF IN READING -- STOPPING EXECUTION*/
     +       * FILE NUMBER*,I2,*   NUMBER OF BLOCKS READ*,I5)
      END
```

FLOW CHART
FOR SUBROUTINE SUMSN



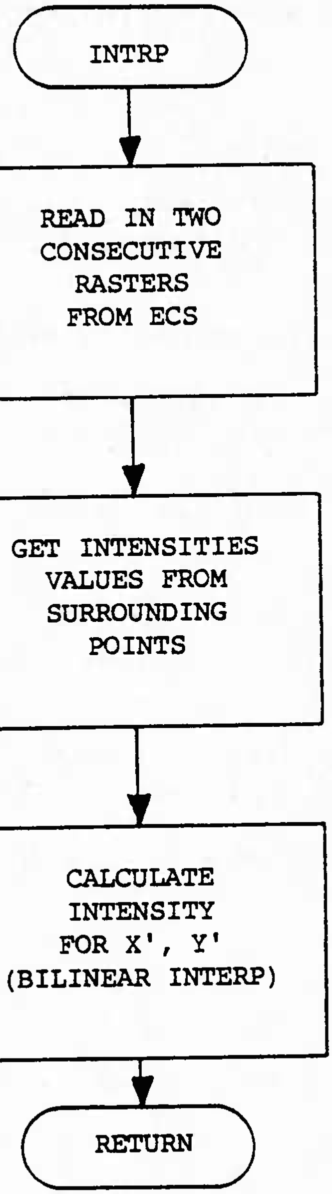
```
      SUBROUTINE SUMSN(NOPTS,NINC,NMAX)
C
C THIS SUBROUTINE CALCULATES THE CROSS-CORRELATION, S(N), WHICH IS
C (THE SUM OF) IMG1(I) TIMES IMG2(I*N) FOR N EQUAL TO +/- NINC*5.
C
C INPUT: NINC, INCREMENT OF PIXELS
C        NMAX, ZERO OR THE NINC WHICH GAVE THE LARGEST CORRELATION
C        NOPTS, NUMBER OF POINTS IN EACH CONVERTED IMAGE ARRAY
C        INTENS, COMMON AREA CONTAINING THE INTENSITIES OF THE CONVERTED
C                PIXELS ON THE ANNULUS.
C OUTPUT: NMAX THE HIGHEST SN VALUE AT THIS N
C
      DIMENSION SN(11)
      COMMON /INTENS/ IMG1(1602), IMG2(1602)
C
10    SNMAX = 0.
      NN = N = -5*NINC + NMAX
      PRINT 111, NINC
111   FORMAT(1H0,35HN VS. S(N), WHICH INCREMENT OF N IS,I3)
      PRINT 112
112   FORMAT(1H0,* N *,T20,*S(N)*/)
C
      DO 100 IN=1,11
        SN(IN) = 0.
        DO 50 J=1,NOPTS
          K = J+N
          IF (K.LT.1) K=K+NOPTS
          IF (K.GT.NOPTS) K=K-NOPTS
          SN(IN) = IMG1(J)*IMG2(K) + SN(IN)
50      CONTINUE
        PRINT 114, N, SN(IN)
        IF (SN(IN).LT.SNMAX) GOTO 80
        SNMAX = SN(IN)
        NMAX = N
80      N = N+NINC
100   CONTINUE
114   FORMAT(1H ,I3,F22.2)
      PRINT 115,NMAX
115   FORMAT(1H0,* N MAXIMUM IS *,I3)
      IF (NN.EQ.NMAX .OR. IABS(NN).EQ.NMAX) GOTO 10
      RETURN
      END
```

FLOW CHART
FOR SUBROUTINE ROTAVG



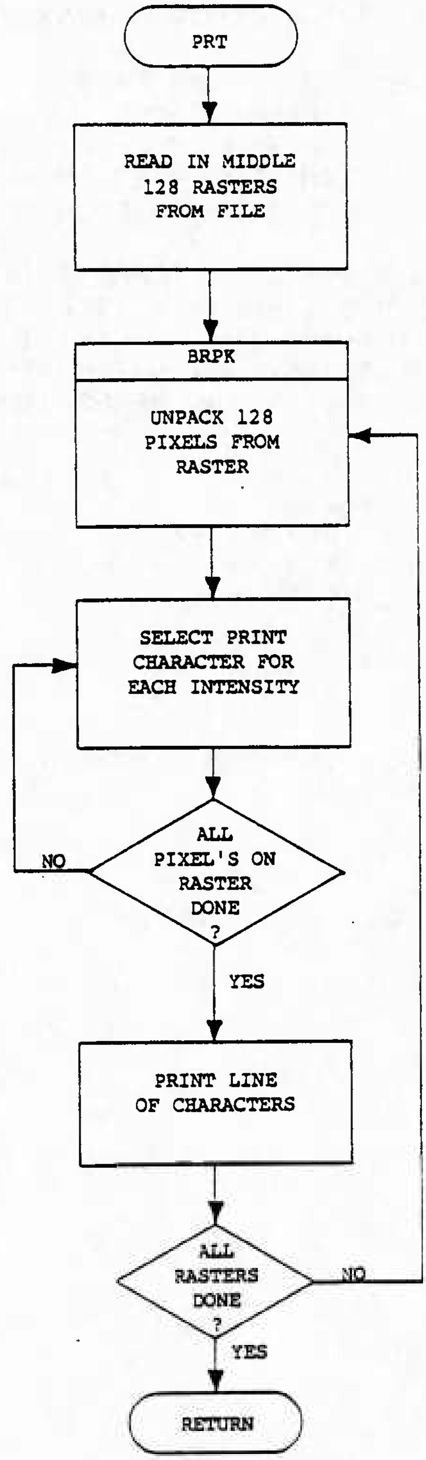
```
      SUBROUTINE ROTAVG(ANGLE)
C
C THIS SUBROUTINE ROTATES IMAGE1 CLOCKWISE AND IMAGE2 COUNTER-
C CLOCKWISE THE SAME AMOUNT.  THE IMAGES ARE THEN AVERAGED TO GET
C THE NEW INTERPOLATED IMAGE.
C
C INPUT:   ANGLE - AMOUNT IN RADIANS TO ROTATE THE IMAGES
C          ECBLK -  FIRST WORD ADDRESSES OF EACH RASTER OF THE IMAGES.
C
C OUTPUT:   THE NEW INTERPOLATED IMAGE
C
      DIMENSION IMG3(69), NEW(512), INT(2), COSANG(2), SINANG(2)
      COMMON /ECBLK/ IEC1(515), IEC2(515), IEC3(512)
C
C- - -INITIALIZE VARIABLES
C
      SINANG(1)  = SIN(ANGLE)
      COSANG(1)  = COS(ANGLE)
      SINANG(2)  = SIN(-ANGLE)
      COSANG(2)  = COS(-ANGLE)
      RR = 255. * 255.
C
C- - -ROTATE IMAGES
C
      DO 200 J = 1,512
C- - -CALCULATE RASTER OF NEW IMAGE, JUST DO A 255.0 RADIUS CIRCLE
        IEC3(J)=69*(J+1029)
        CALL ZILCH(NEW,512)
        Y = J - 1
        IF (Y.EQ.511.) GOTO 50
        YOLD = 255.0 - Y
        X2X2 = RR - (YOLD)*(YOLD)
        IF (X2X2.LT.0.) GOTO 50
        X2 = SQRT(X2X2) + 255.0
        I2 = X2+1.
        I1 = 511.5 - X2
        IF (I2.GT.I1) GOTO 100
   50     CALL WRITEC(NEW,IEC3(J),69)
          GOTO 200
C- - -INTERPOLATE ONE RASTER OF EACH IMAGE
  100   DO 150 I=I1,I2
          XOLD = I - 256
          DO 125 IM=1,2
            XPRIME = (XOLD*COSANG(IM)) + (YOLD*SINANG(IM))
            YPRIME = (YOLD*COSANG(IM)) - (XOLD*SINANG(IM))
            X = XPRIME + 255.0
            Y = 255.0 - YPRIME
            IF (X.LT.1.0 .OR. X.GE.512.0) GOTO 150
            IF (Y.LT.1.0 .OR. X.GE.512.0) GOTO 150
            CALL INTRP(IM,X,Y,INT(IM))
  125     CONTINUE
C- - -AVERAGE PIXELS
          IF (INT(1).EQ.0) INT(2)=0
          IF (INT(2).EQ.0) INT(1)=0
          NEW(I) = FLOAT(INT(1)+INT(2)) * 0.5 +.5
  150   CONTINUE
C- - -PACK RASTER (512 PIXELS INTO 69 WORDS) AND STORE RASTER.
        CALL BRPK(NEW,60,1,IMG3,8,1,512)
        CALL WRITEC(IMG3,IEC3(J),69)
  200 CONTINUE
      RETURN
      END
```

FLOW CHART
OF SUBROUTINE INTRP



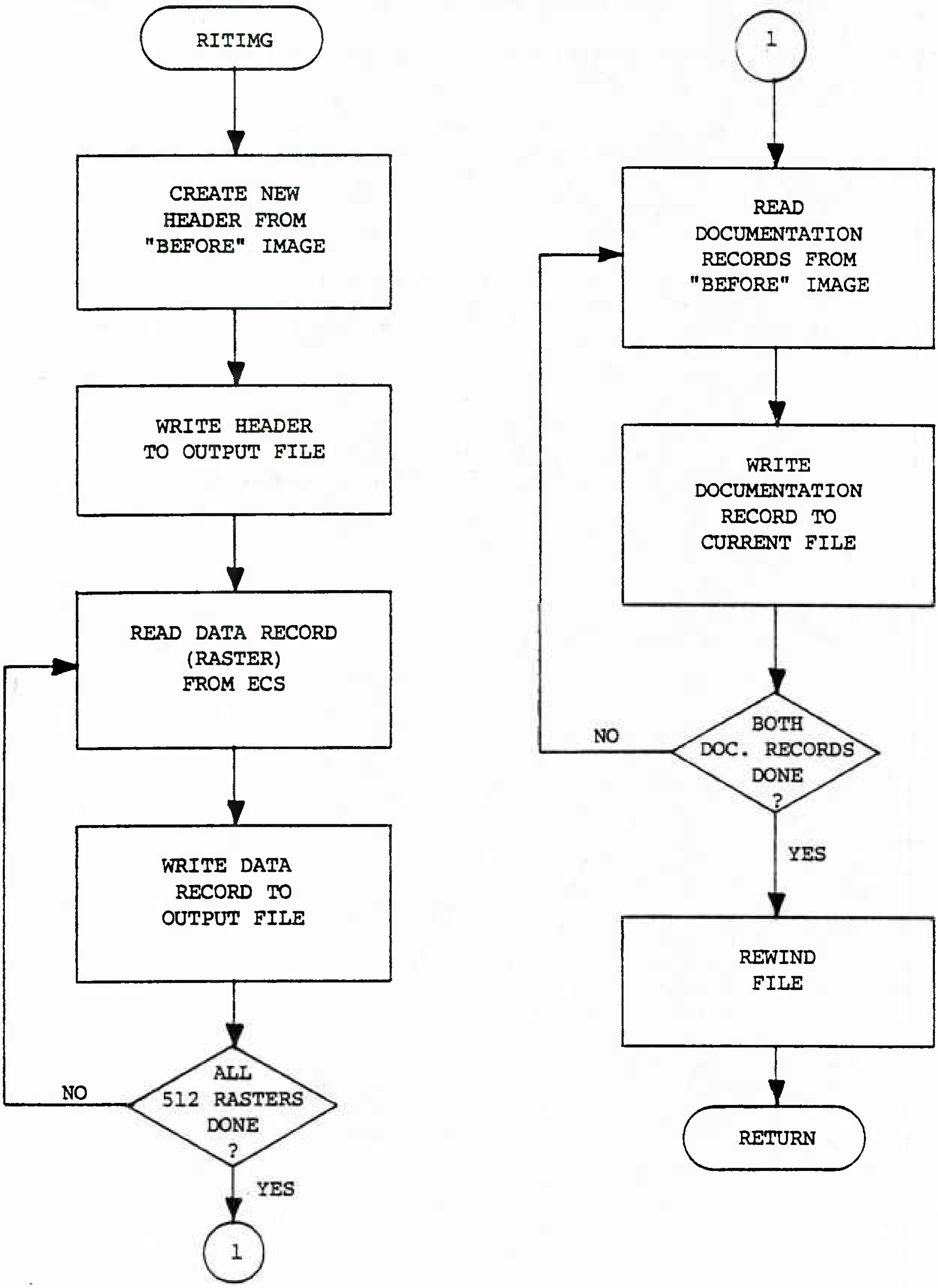
```
      SUBROUTINE INTRP(IFILE,X,Y,INT)
C
C THIS SUBROUTINE DOES A BILINEAR INTERPOLATION TO GET THE INTENSITY
C VALUE AT POINT(X,Y)
C
C INPUT:
C    IFILE    - FILE TO READ RASTERS FROM
C    X, Y     - LOCATION OF POINT
C               FROM TOP-LEFT CORNER OF IMAGE, BEING (1,1)
C    ECBLK    - COMMON BLOCK WITH STORAGE LOCATIONS OF EACH RASTER
C
C OUTPUT:
C    INT      - INTERPOLATED INTENSITY VALUE AT (X,Y)
C
      DIMENSION IR1(512), IR2(512), IDATA(138)
      COMMON /ECBLK/ IEC1(515),IEC2(515),IEC3(512)
C
C- - -FIND LOCATION OF INTENSITY VALUES
C
      IY = IFIX(Y)
      IX = IFIX(X)
      IF (IFILE.EQ.2) GOTO 5
      CALL READEC(IDATA,IEC1(IY),138)
      GOTO 10
5     CALL READEC(IDATA,IEC2(IY),138)
10    CALL BRPK(IDATA,8,1,IR1,60,1,512)
      CALL BRPK(IDATA(70),8,1,IR2,60,1,512)
C
C- - -GET INTENSITIES
C
      P0 = IR1(IX)
      P1 = IR1(IX+1)
      P2 = IR2(IX)
      P3 = IR2(IX+1)
C
C- - -CALCULATE INTENSITY AT POINT(X,Y) FROM SURROUNDING POINTS
C
      DX = X - FLOAT(IX)
      DY = Y - FLOAT(IFIX(Y))
      XINT = (P1-P0)*DX + (P2-P0)*DY + (P3+P0-P2-P1)*DX*DY + P0
      INT = XINT + 0.5
      RETURN
      END
```

FLOW CHART
FOR SUBROUTINE PRT



```
      SUBROUTINE PRT(IMGFILE)
C
C THIS SUBROUTINE PRINTS OUT THE CENTRAL 128X128 PORTION OF AN IMAGE.
C
C INPUT:   IMGFILE - UNIT NUMBER OF THE FILE
C                    UNIT 1 - BEFORE IMAGE
C                         2 - AFTER IMAGE
C                         3 - INTERPOLATED IMAGE
C        ECBLK    - STORAGE LOCATION OF EACH RASTER
C
      DIMENSION IMAGE(128), IM(128), ICHAR(16), NAME(3), IFILE(8832)
      COMMON /ECBLK/ IEC1(515),IEC2(515),IEC3(512)
      DATA ICHAR/1H ,1H.,1H:,1H-,1H+,1H*,1H1,1H2,
     +           1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H=/
      DATA NAME/10HTHE BEFORE,10H THE AFTER,10H   THE NEW/
C
C
      PRINT 1, NAME(IMGFILE)
    1 FORMAT(1H1,A10,* IMAGE PRINTED*)
      IF (IMGFILE.EQ.1) IEC=IEC1(193)
      IF (IMGFILE.EQ.2) IEC=IEC2(193)
      IF (IMGFILE.EQ.3) IEC=IEC3(193)
      CALL READEC(IFILE,IEC,8832)
      L=1
      DO 400 J=193,320
        CALL BRPK(IFILE(L),8,193,IMAGE,60,1,128)
        DO 300 I=1,128
          K = I
          INT = IMAGE(I)
          INT = INT/16 + 1
          IM(K) = ICHAR(INT)
  300     CONTINUE
        PRINT 2, (IM(K),K=1,128)
        L=L+69
  400 CONTINUE
    2 FORMAT(1H ,128(A1))
      RETURN
      END
```

FLOW CHART
FOR SUBROUTINE RITIMG



```
      SUBROUTINE RITING
C THIS SUBROUTINE WRITES THE NEWLY CREATED IMAGE TO THE FILE, TAPE5.
C INPUT:
C    ECBLK - COMMON BLOCK WITH STORAGE LOCATIONS OF EACH RASTER
C            IEC1 - BEFORE IMAGE
C            IEC2 - AFTER IMAGE
C            IEC3 - NEW INTERPOLATED IMAGE
C OUTPUT: THE NEW INTERPOLATED IMAGE WRITTEN TO TAPE OR DISK FILE.
C         THE FORMAT OF THE FILE WILL BE:
C            RECORD NO.    CONTENTS
C            ---------     --------
C                1         HEADER
C              2 - 513     DATA (PIXEL INTENSITIES)
C            514 - 515     DOCUMENTATION
C         THE HEADER AND DOCUMENTATION RECORDS WILL BE COPIED FROM
C         THE BEFORE IMAGE.
C
C
      DIMENSION IMAGE(69), IHEADER(69), IWK(8)
      COMMON /ECBLK/ IEC1(515), IEC2(515), IEC3(512)
      DATA IWK/10HIMAGE*INTE,10HRPOLATED*A,10HFTER******,5*0/
C
C- - -INITIALIZE
C
      MSK1 = 1001001001001001001001B
      MSK2 = 0100100100100100100100B
      MSK3 = 0010010010010010010010B
C
C- - -HEADER RECORD
C
      CALL BRPK(IWK,6,1,IWK,9,1,30)
      IWK(1) = IWK(1) .OR. MSK1
      IWK(2) = IWK(2) .OR. MSK2
      IWK(3) = IWK(3) .OR. MSK3
      IWK(4) = IWK(4) .OR. MSK1
      CALL BRPK(IWK,9,1,IHEADER,8,1,30)
      CALL READEC(IMAGE,IEC1(515),69)
      IHEADER(4) = SHIFT( IMAGE(1).AND.MASK(56),+56)
      ICHAR = IMAGE(1).AND..NOT.MASK(56)
      ISCRTCH = IMAGE(2).AND.MASK(56).OR.ICHAR
      IHEADER(5)= SHIFT(ISCRTCH,+56)
      CALL BRPK(IHEADER,8,1,IWK,12,1,80)
      CALL ASCIDPC(IWK,16,IMAGE,16,LDPC,IERR)
      IF (IERR.EQ.1) PRINT 222,LDPC
      PRINT 333,(IMAGE(L),L=1,LDPC)
      IHEADER(5) = IMAGE(2)
      BUFFER OUT (5,1) (IHEADER(1),IHEADER(69))
      IF (UNIT(5)) 20,110,110
```

```
C
C- - -WRITE DATA AND DOCUMENTATION RECORDS
C
   20 DO 30 N=2,515
        LOC = N-1
        IF (LOC.GT.512) GOTO 25
        CALL READEC(IMAGE,IEC3(LOC),69)
        GOTO 27
   25   CALL READEC(IMAGE,IEC1(LOC),69)
   27   BUFFER OUT (5,1) (IMAGE(1),IMAGE(69))
        IF (UNIT(5)) 30,110,110
   30 CONTINUE
C
      ENDFILE 5
      REWIND 5
      RETURN
C
C- - -ERROR EXIT
C
  110 PRINT 111,N
  111 FORMAT(* ERROR -- EOF OR PARITY IN RITING, NO. BLOCKS = *,I5)
  222 FORMAT(* ERROR TOO SMALL IMAGE ARRAY, LDPC =*,I5)
  333 FORMAT(* NEW IMAGE HEADER IS *,8A10)
C
      RETURN
      END
```

APPENDIX D

Listing of Abbreviations and Definitions used in this report

ABBREVIATIONS AND DEFINITIONS

CDC	Control Data Corporation
CM	Central Memory
DG	Data General
ECS	Extended Core Storage
FNOC	Fleet Numerical Oceanography Center
GADHS	GOES Acquisition and Data Handling System
GOES	Geostationary Operational Environmental Satellite
HAL	A computer at FNOC (a CDC 6500)
MCIDAS	Man Computer Interactive Data Access System
NEPRF	Naval Environmental Prediction Research Facility
NTSC	National Television System Committee
ODSI	Ocean Data Systems, Inc.
PGP	Programmable Graphic Processor, part of the Genisco GCT-3000 display unit
Pixel	Picture element, the smallest addressable element of an image
Raster	A row of pixels
RBG	Red-Blue-Green, the basic colors from which digital images are built
SPADS	Satellite-data Processing and Display System
SPC	A computer at FNOC (a CDC CY175)
VSN	Volume Serial Number, for magnetic tapes

DISTRIBUTION

COMSECONDFLT
ATTN: NSAP SCIENCE ADVISOR
FPO NEW YORK 09501

COMSEVENTHFLT
FLT METEOROLOGIST, N30W
FPO SAN FRANCISCO 96601

COMTHIRDFLT
NSAP SCIENCE ADVISOR
CODE N702/01T
PEARL HARBOR, HI 96860

CHIEF OF NAVAL RESEARCH (2)
LIBRARY SERV., CODE 734
RM 633, BALLSTON TOWER #1
800 QUINCY ST.
ARLINGTON, VA 22217

OFFICE OF NAVAL RESEARCH
NARAD BRIEFING FILM REPORT
800 N. QUINCY ST.
ARLINGTON, VA 22217

NAVAL DEPUTY TO THE
ADMINISTRATOR, NOAA
RM 200, PAGE BLDG. #1
3300 WHITEHAVEN ST. NW
WASHINGTON, DC 20235

OFFICER IN CHARGE
NAVOCEANCOMDET
MONTEREY, CA 93940

LIBRARY
NAVAL ARCTIC RESEARCH LAB
BARROW, AK 99723

COMMANDING OFFICER
NAVAL RESEARCH LAB
ATTN: LIBRARY, CODE 2620
WASHINGTON, DC 20390

COMMANDING OFFICER
NAVOCEANO LIBRARY
NSTL STATION
BAY ST. LOUIS, MS 39522

COMMANDING OFFICER
FLENUMOCEANCEN
MONTEREY, CA 93940

COMMANDING OFFICER
NAVWESTOCEANCEN
BOX 113
PEARL HARBOR, HI 96860

COMMANDING OFFICER
NAVEASTOCEANCEN, NAS
MCADIE BLDG. (U-117)
NORFOLK, VA 23511

COMMANDING OFFICER
NAVPOLAROCEANCEN
NAVY DEPT.
4301 SUITLAND RD.
WASHINGTON, DC 20390

COMMANDING OFFICER
U.S. NAVOCEANCOMCEN
BOX 12, COMNAVMARIANAS
FPO SAN FRANCISCO 96630

COMMANDING OFFICER
U.S. NAVOCEANCOMCEN
BOX 31 (ROTA)
FPO NEW YORK 09540

SUPERINTENDENT
LIBRARY REPORTS
U.S. NAVAL ACADEMY
ANNAPOLIS, MD 21402

PRESIDENT
NAVAL WAR COLLEGE
ATTN: GEOPHYSICS OFFICER
NEWPORT, RI 02840

COMMANDER (2)
NAVAIRSYSCOM
ATTN: LIBRARY, AIR-00D4
WASHINGTON, DC 20361

COMMANDER
NAVAIRSYSCOM, AIR-33
WASHINGTON, DC 20361

COMMANDER
PACMISTESTCEN
GEOPHYSICS OFFICER
CODE 3250
PT. MUGU, CA 93042

NAVAL POSTGRADUATE SCHOOL
METEORO. DEPT., CODE 63
MONTEREY, CA 93940

NAVAL POSTGRADUATE SCHOOL
ATTN: LIBRARY
MONTEREY, CA 93940

USAFETAC/TS
SCOTT AFB, IL 62225

3350TH TECH. TRNG GROUP
TTGU/2/STOP 623
CHANUTE AFB, IL 61868

AFGWC/DAPL
OFFUTT AFB, NE 68113

AFGL/LY
HANSCOM AFB, MA 01731

AFGL/OPI
HANSCOM AFB, MA 01731

5WW/DN
LANGLEY AFB, VA 23665

CHIEF
AEROSPACE SCIENCE BRANCH
HQ 1ST WW (MAC)
HICKAM AFB, HI 96853

COMMANDER & DIRECTOR
ATTN: DELAS-DM-A
U.S. ARMY ATMOS. SCI. LAB
WHITE SANDS MISSILE RANGE
WHITE SANDS, NM 88002

DIRECTOR
LIBRARY, TECH. INFO. CEN.
ARMY ENGS. WATERWAYS STN.
VICKSBURG, MS 39180

DIRECTOR (12)
DEFENSE TECH. INFORMATION
CENTER, CAMERON STATION
ALEXANDRIA, VA 22314

CHIEF
MARINE SCIENCE SECTION
U.S. COAST GUARD ACADEMY
NEW LONDON, CT 06320

DIRECTOR
OFFICE OF PROGRAMS RX3
NOAA RESEARCH LAB
BOULDER, CO 80302

DIRECTOR, NATIONAL EARTH
SATELLITE SERV/SEL
FB-4, S321B
SUITLAND, MD 20233

DIRECTOR
NATIONAL HURRICANE CENTER
NOAA, GABLES ONE TOWER
1320 S. DIXIE HWY
CORAL GABLES, FL 33146

CHIEF, SCIENTIFIC SERVICES
NWS, PACIFIC REGION
P.O. BOX 50027
HONOLULU, HI 96850

NOAA RESEARCH FACILITIES
CENTER
P.O. BOX 520197
MIAMI, FL 33152

DIRECTOR, CENTRAL PACIFIC
HURRICANE CENTER
NWS, NOAA
HONOLULU, HI 96819

CHIEF
MESOSCALE APPL. BRANCH
NATIONAL EARTH SAT. SERV.
1225 W. DAYTON
MADISON, WI 53562

MANAGER
NATIONAL EARTH SAT. SERV.
SATELLITE FIELD SERV. STN.
701 C. ST., BOX 45
ANCHORAGE, AK 99513

DIRECTOR (AOML)
NATIONAL HURRI. RSCH. LAB.
1320 S. DIXIE HWY.
CORAL GABLES, FL 33146

DIRECTOR
TECHNIQUES DEVELOPMENT LAB
GRAMAX BLDG.
8060 13TH ST.
SILVER SPRING, MD 20910

NATIONAL WEATHER SERV./NOAA
ATTN: J. W. NICKERSON
MARINE OBSERV. PROG. LEADER
GRAMAX BLDG. 8060 13TH ST.
SILVER SPRING, MD 20910

THE EXECUTIVE DIRECTOR
AMERICAN METEORO. SOCIETY
45 BEACON ST.
BOSTON, MA 02108

AMERICAN METEORO. SOCIETY
METEORO/GEOASTRO ABSTRACTS
P.O. BOX 1736
WASHINGTON, DC 20013

LAB. FOR ATMOS. SCIENCES
NASA
GODDARD SPACE FLIGHT CENTER
GREENBELT, MD 20771

SCRIPPS INSTITUTION OF
OCEANOGRAPHY, LIBRARY
DOCUMENTS/REPORTS SECTION
LA JOLLA, CA 92037

DIRECTOR
CENTER FOR MARINE STUDIES
SAN DIEGO STATE UNIVERSITY
SAN DIEGO, CA 92182

CHAIRMAN, METEOROLOGY DEPT.
CALIFORNIA STATE UNIVERSITY
SAN JOSE, CA 95192

COLORADO STATE UNIVERSITY
ATMOS. SCIENCES DEPT.
ATTN: DR. ROGER PIELKE
FT. COLLINS, CO 80523

DIRECTOR, REMOTE SENSING LAB
UNIVERSITY OF MIAMI
P.O. BOX 248003
CORAL GABLES, FL 33124

CHAIRMAN, METEOROLOGY &
PHYSICS DEPT.
UNIVERSITY OF FLORIDA
215 PHYSICS BLDG.
GAINESVILLE, FL 32601

FLORIDA STATE UNIVERSITY
ENVIRONMENTAL SCI. DEPT.
TALLAHASSEE, FL 32306

UNIVERSITY OF HAWAII
METEOROLOGY DEPT.
2525 CORREA ROAD
HONOLULU, HI 96822

UNIVERSITY OF CHICAGO
ATMOSPHERIC SCIENCES DEPT.
1100 E. 57TH ST.
CHICAGO, IL 60637

DIRECTOR
COASTAL STUDIES INSTITUTE
LOUISIANA STATE UNIVERSITY
ATTN: O. HUH
BATON ROUGE, LA 70803

DIRECTOR, JTWC
BOX 17
FPO SAN FRANCISCO 96630

ACQUISITION SET. IRDB-D823
LIBRARY & INFO. SERV., NOAA
6009 EXECUIVE BLVD.
ROCKVILLE, MD 20852

UNIVERSITY OF MARYLAND
METEOROLOGY DEPT.
COLLEGE PARK, MD 20742

CHAIRMAN, METEOROLOGY DEPT.
MASSACHUSETTS INSTITUTE OF
TECHNOLOGY
CAMBRIDGE, MA 02139

CHAIRMAN, ATMOSPHERIC
SCIENCE DEPT.
UNIV. OF MISSOURI, COLUMBIA
701 HITT ST.
COLUMBIA, MO 65211

DIRECTOR OF METEOROLOGY
EARTH & ATMOS. SCI. DEPT.
ST. LOUIS UNIVERSITY
P.O. BOX 8099
ST. LOUIS, MO 63156

CHAIRMAN, METEOROLOGY DEPT.
PENNSYLVANIA STATE UNIV.
503 DEIKE BLDG.
UNIVERSITY PARK, PA 16802

TEXAS A&M UNIVERSITY
METEOROLOGY DEPT.
COLLEGE STATION, TX 77843

CHAIRMAN
ATMOS. SCIENCES DEPT.
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VA 22903

UNIVERSITY OF WASHINGTON
ATMOSPHERIC SCIENCES DEPT.
SEATTLE, WA 98195

CHAIRMAN, METEOROLOGY DEPT.
UNIVERSITY OF WISCONSIN
1225 W. DAYTON ST.
MADISON, WI 53706

COLORADO STATE UNIVERSITY
ATMOSPHERIC SCIENCES DEPT.
ATTN: LIBRARIAN
FT. COLLINS, CO 80521

SAINT LOUIS UNIVERSITY
EARTH & ATMOS. SCI. DEPT.
BOX 8099-LACLEDE STATION
ST. LOUIS, MO 63156

WORLD METEOROLOGICAL
ORGANIZATION, ATS DIV.
ATTN: N. SUZUKI
CH-1211, GENEVA 20
SWITZERLAND

BUREAU OF METEOROLOGY
ATTN: LIBRARY
BOX 1289K, GPO
MELBOURNE, VIC, 3001
AUSTRALIA